

■ PROJECTS ■ THEORY ■ APPLICATIONS ■ CIRCUITS ■ TECHNOLOGY

NUTS
AND
VOLTS

NUTS AND VOLTS

www.nutsvolts.com
December 2014

EVERYTHING FOR ELECTRONICS

Holiday Line-up

3D LED Christmas Tree

Brighten things up with your own home-built tree

Synchronized Light Displays

Creative tips and overview of a basic system

Make Your Own SMT Earrings

The ultimate in holiday bling

Eight-Channel Light Controller

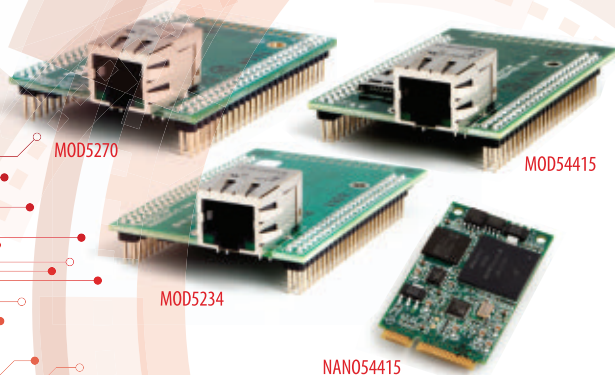
Use random numbers to make sure your light displays never look the same



New!

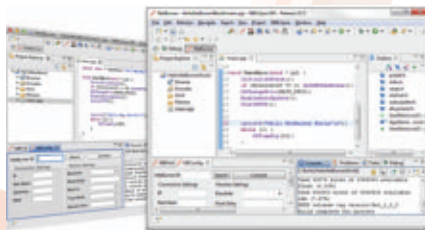
Getting Started With 3D Printing

Ethernet Core Modules with High-Performance Connectivity Options



- **MOD5270**
147.5 MHz processor with 512KB Flash & 8MB RAM · 47 GPIO · 3 UARTs · I²C · SPI
- **MOD5234**
147.5 MHz processor with 2MB flash & 8MB RAM · 49 GPIO · 3 UARTs · I²C · SPI · CAN · eTPU (for I/O handling, serial communications, motor/timing/engine control applications)
- **MOD54415**
250 MHz processor with 32MB flash & 64MB RAM · 42 GPIO · 8 UARTs · 5 I²C · 3 SPI · 2 CAN · SSI · 8 ADC · 2 DAC · 8 PWM · 1-Wire® interface
- **NANO54415**
250 MHz processor with 8MB flash & 64MB RAM · 30 GPIO · 8 UARTs · 4 I²C · 3 SPI · 2 CAN · SSI · 6 ADC · 2 DAC · 8 PWM · 1-Wire® interface

Add Ethernet connectivity to an existing product, or use it as your product's core processor



The goal: Control, configure, or monitor a device using Ethernet

The method: Create and deploy applications from your Mac or Windows PC. Get hands-on familiarity with the NetBurner platform by studying, building, and modifying source code examples.

The result: Access device from the Internet or a local area network (LAN)

The NetBurner Ethernet Core Module is a device containing everything needed for design engineers to add network control and to monitor a company's communications assets. For a very low price point, this module solves the problem of network-enabling devices with 10/100 Ethernet, including those requiring digital, analog and serial control.

MOD5270-100IR.....\$69 (qty. 100)	NNDK-MOD5270LC-KIT\$99
MOD5234-100IR.....\$99 (qty. 100)	NNDK-MOD5234LC-KIT\$249
MOD54415-100IR.....\$89 (qty. 100)	NNDK-MOD54415LC-KIT\$129
NANO54415-200IR...\$69 (qty. 100)	NNDK-NANO54415-KIT.....\$99

NetBurner Development Kits are available to customize any aspect of operation including web pages, data filtering, or custom network applications. The kits include all the hardware and software you need to build your embedded application.

➤ **For additional information please visit**
<http://www.netburner.com/kits>

all the inventions waiting
to happen...

CLICKER² PIC18FJ

price:
\$39⁰⁰

Come See the Always-Open Classroom!

LEARN.DIGILENTINC.COM

Learn.digilentinc.com is an open, freely accessible community dedicated to providing teaching and learning materials focused on the electronics skills needed by beginners and practicing engineers alike. The site is built around a collection of hands-on projects that offer insight and design experience in key areas like analog circuit design, digital circuit design, and microprocessor programming. Each project is self-contained and designed to expose a fundamental concept in hardware or software design. You can go through each project at your own pace or take the time to explore related topics in greater detail. Use these materials to learn the fundamentals you need to take your projects to the next level.

Come visit the always-open classroom today!

Ohm's Law

In 1827, George Ohm demonstrated through a series of experiments that voltage, current, and resistance are related through a fundamental relationship. Voltage (V) is equal to Current (I) times resistance (R), or $V = I \cdot R$. This most basic equation in electronics shows that when any two of the three quantities are known, the third can be derived. The Ohm's law triangle in Fig. 1 below shows how to derive each variable given the other two. An important thing to note about Ohm's law is that a resistor's voltage and current are related by its resistance.

$$V = I \cdot R$$

Getting Started with FPGA

Project 1: Introduction to Digital Engineering and FPGA Board

Introduction

This project sets up your FPGA board for use and shows you the steps in starting project files. It also gives you some basic background knowledge on electrical engineering and digital circuits. This is a starter project with very little hands-on work with your board, but it is a good reference if you ever forget how to start your projects.

Before you begin, you should:

- Have a Digilent® FPGA board.
- Have a PC with internet access.
- Have some time to set up your board and develop the tools for further projects.

After you're done, you should:

- Have the Digilent® ISE WebPACK™ installed.
- Have a basic understanding of what digital engineering is.
- Have time to set up a "Hello World" project for your FPGA board.

Inventory:

Qty	Description
1	Digilent-PRO3000, PRO3000L or PRO3000LX FPGA Board
1	Wires (100-ohm termination: Protektor 0.1 or 0.2 recommended)
1	Digilent adapter

Basic Theory

The use of the word circuit in this context comes from the idea that electric power must flow from the positive terminal of a power source through one or more electronic devices and back to the negative terminal of a power source, thereby forming a circuit. If the connections between an electronic device and either the positive or negative terminals of a power supply are interrupted, the circuit will be broken and the device will not function.

A digital circuit consists of a power supply, drivers, and collection nets. Some nets provide circuit inputs from the "outside world"; in a schematic, these input nets are generally shown entering the left side of component and/or the overall circuit. Other nets present circuit outputs to the outside world; these nets are generally shown exiting the schematic on the right side. Current components are shown in arbitrary shapes; nets are shown as lines, and inputs and outputs are denoted by connector symbols.

In a digital circuit, power supply voltage levels are constrained to two distinct values – "logic high voltage" (called LV_H or V_{H0}) and "logic low voltage" (called LV_L or V_{L0}). The GND node in any circuit is the universal reference voltage against which all other voltages are measured (in modern digital circuits, GND is typically the lowest voltage in the circuit). In a schematic, it is often difficult to show base connections of circuit modules, so an assumed connection is shown to the common GND node.

Related Material

- Electronic Charge, Voltage & Current
- Resistor & Ohm's Law
- Logic & No. Thresholds
- Electronic Components
- Physical V_{H0} Models

LIKE COAL. BUT FASTER.



ollie

One part coal. One part extreme. This is **Darkside Ollie** - the naughtiest app-controlled robot ever created. Rocket around at a floor-warping 14 MPH, pull off diabolical tricks, and smoke the competition. You can find Darkside Ollie at the top of the naughty list - and sold exclusively at gosphero.com.

It's time to upgrade your play.

gosphero.com



December 2014

26 Synchronized Light Displays

Ready to give Clark Griswold a run for his money with your next holiday display? Then, you'll appreciate these tips on the creative process and an overview of putting together a basic lighting system.

■ By John Storms

32 Build the 3D Christmas LED Tree

Brighten things up this holiday season with your own home-built LED tree.

■ By Ron Newton

38 Make These Unique Xmas Earrings

If an LED Christmas tree is simply not enough bling this year, then you'll want to whip up a bunch of these sparkly earring sets to make things even more festive. Plus, they make great gifts.

■ By Ron Newton

44 Wishing You a Very Random Christmas

This flexible eight-channel light controller uses random number generators to make sure your holiday displays never look the same.

■ By Dan Koellen

51 **PRODUCT REVIEW** Boxed Kit Amps' Gobo Stereo Amplifier

This is a fun to build desktop stereo amplifier kit that is appropriate for any experience level.

■ By J.W. Koebel



Columns

10 TechKnowledgey 2014 Events, Advances, and News

Topics of interest this time include the world's fastest electric motorcycle, getting your own cloud, taking your own ECG, preserving the CRT, plus some other cool stuff.

14 Q&A Reader Questions Answered Here

Questions cover a battery monitor circuit, NiMHs over NiCads, and using an audio synthesizer IC with an eight ohm speaker.

18 The Spin Zone Adventures in Propeller Programming

Putting a Button On It.

Think there's not a lot to say about implementing buttons into your electronic projects? Guess again.

57 Smiley's Workshop Programming • Hardware • Projects The Arduino Classroom.

Arduino 101/Chapter 11: Sound.

In order to make sounds using the Arduino, you need to know about arrays and interrupts.

68 Getting Started With 3D Printing **NEW!** Helpful Instructions to Create and Print

Bridging the Gap of Packaging With 3D Printing.

The three main areas to designing a project are: hardware, software, and packaging. While hardware and software have come a long way over the years, replicating custom enclosures has been on the expensive side ... until now.

72 Near Space Approaching the Final Frontier CubSats — Part 1.

There's a new breed of satellites in town with a lot of potential to cost-effectively and boldly go where no man has before.

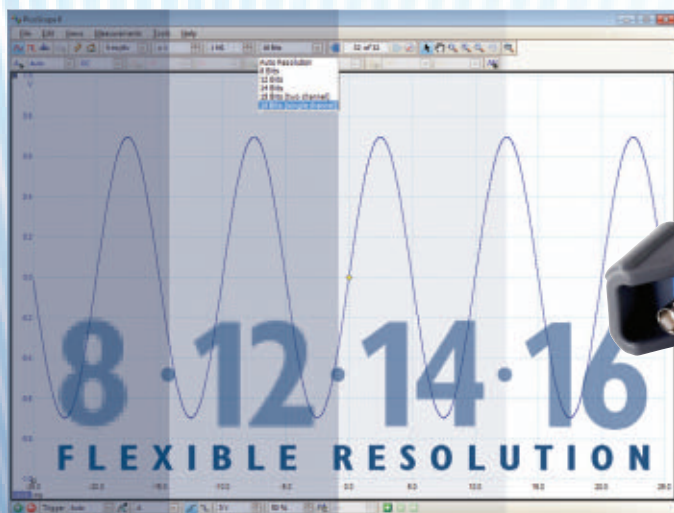


Departments

08 DEVELOPING PERSPECTIVES
The Art of Prototyping
24 NEW PRODUCTS
25 SHOWCASE
66 ELECTRO-NET

75 CLASSIFIEDS
76 NV WEBSTORE
79 TECH FORUM
81 AD INDEX

WHY COMPROMISE SPEED v ACCURACY? HAVE IT ALL

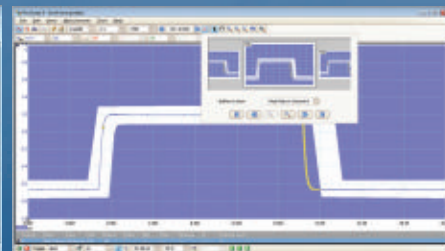
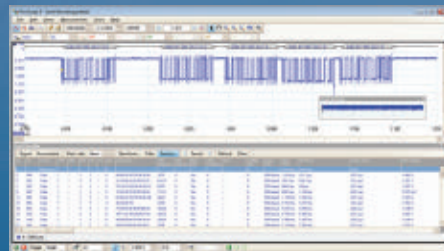
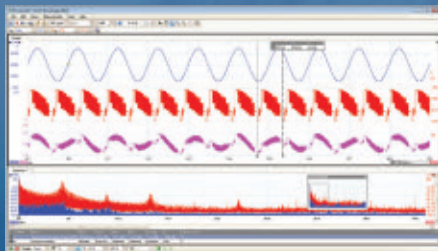


PicoScope	PicoScope 5442A	PicoScope 5442B	PicoScope 5443A	PicoScope 5443B	PicoScope 5444A	PicoScope 5444B
Channels	4					
Bandwidth	All modes: 60 MHz		8 to 15-bit modes: 100 MHz 16-bit mode: 60 MHz		8 to 15-bit modes: 200 MHz 16-bit mode: 60 MHz	
Max. sampling rate	8-bit mode		12-bit mode	14-bit mode	15-bit mode	16-bit mode
Any 1 channel	1 GS/s		500 MS/s	125 MS/s	125 MS/s	125 MS/s
Any 2 channels	500 MS/s		250 MS/s	125 MS/s	125 MS/s	-
Any 3 channels	250 MS/s		125 MS/s	125 MS/s	-	-
Four channels	250 MS/s		125 MS/s	125 MS/s	-	-
Sampling rate - ETS (8-bit mode only)	2.5 GS/s		5 GS/s		10 GS/s	
Buffer memory (8-bit) *	16 MS	32 MS	64 MS	128 MS	256 MS	512 MS
Buffer memory (≥ 12 -bit)*	8 MS	16 MS	32 MS	64 MS	128 MS	256 MS
Resolution (enhanced)**	8 bits, 12 bits, 14 bits, 15 bits, 16 bits (hardware resolution + 4 bits)					
Signal Generator	Function generator	AWG	Function generator	AWG	Function generator	AWG

2 Channel models also available * Shared between active channels

** Maximum resolution is limited on the lowest voltage ranges: ± 10 mV = 8 bits • ± 20 mV = 12 bits. All other ranges can use full resolution.

FLEXIBLE RESOLUTION OSCILLOSCOPE



ALL MODELS INCLUDE PROBES, FULL SOFTWARE AND 5 YEAR WARRANTY. SOFTWARE INCLUDES MEASUREMENTS, SPECTRUM ANALYZER, SDK, ADVANCED TRIGGERS, COLOR PERSISTENCE, SERIAL DECODING (CAN, LIN, RS232, I²C, I²S, FLEXRAY, SPI), MASKS, MATH CHANNELS, ALL AS STANDARD, WITH FREE UPDATES.

www.picotech.com/PCO532 CALL TOLL FREE: 1-800-591-2796



by Bryan Bergeron, Editor

DEVELOPING PERSPECTIVES

The Art of Prototyping

Prototyping — the process of creating an initial circuit or electromechanical device that's intended to evolve — is an art. It's as much a mindset as it is a collection of techniques and approaches to design. Whether you prototype with breadboards, wire wrap, or circuit boards, you have to approach the work with the expectation that all — if not most — of your work will be discarded in subsequent (and hopefully better) generations of the device.

My favorite toolset for prototyping is wire wrap for circuits and 3D printing in PLA for the supporting infrastructure. I use wire wrap IC sockets for both ICs and leaded components, a wire wrap gun, and a handheld unwrapping tool. The unwrapping tool — which is used to remove wrapping from a wire wrap post — is indispensable. 3D printing — especially in PLA — is fast, clean, and relatively inexpensive. There's no heated platform and no plastic tape to worry about, and unlike ABS plastic printing, there's minimal setup between prints. (Be sure to check out the new

column, *Getting Started With 3D Printing* in this issue.)

It's important to distinguish a prototype from the process of prototyping. Some prototypes look like finished products, but they aren't. By finished product, I mean something that can survive shipping, occasional rough handling, and modest operator error such as plugging in the power to a data port. Some prototypes look like final products, but looks are often deceiving.

For example, a lot of the projects featured in *N&V* are prototypes. They may be presented with circuit board files and nice engraved cases, but they're prototypes nonetheless. Kits are typically past the prototype stage in that they have gone through several iterations based on user feedback. However, it's useful to think of kits as prototypes, as well. Even the archetypical electronic kit supplier, HeathKit sold kits that lent themselves to iterative improvement.

So, how do you approach a circuit described in *Nuts & Volts* or a kit advertised in the magazine as a prototype?

The Convenient All-in-One Solution for Custom-Designed Front Panels & Enclosures



FREE Software



ONLY \$90.24
with custom
logo engraving

You design it
to your specifications using
our FREE CAD software,
Front Panel Designer

We machine it
and ship to you a
professionally finished product,
no minimum quantity required

- Cost effective prototypes and production runs with no setup charges
- Powder-coated and anodized finishes in various colors
- Select from aluminum, acrylic or provide your own material
- Standard lead time in 5 days or express manufacturing in 3 or 1 days

**FRONT PANEL
EXPRESS**

FrontPanelExpress.com
1(800)FPE-9060

25 YEARS Beta
LAYOUT
create : electronics

SPECIAL PRICE
to celebrate our anniversary!



\$575.00*

**Complete SMD Workstation:
Anniversary Reflow Kit V3**

save over \$50!*

**Visit us:
electronica 2014
Hall A2, Booth 357
Messe München**

**** as compared with
purchase of the
individual components**

Toll free: 888-977-7443

www.beta-eSTORE.com

eSTORE
Beta LAYOUT

* ships from the U.S.
Shipping costs
not included

NUTS AND VOLTS

EVERYTHING FOR ELECTRONICS

Published Monthly By

T & L Publications, Inc.

430 Princeland Ct.

Corona, CA 92879-1300

(951) 371-8497

FAX (951) 371-3052

Webstore orders only 1-800-783-4624

www.nutsvolts.com

Subscription Orders

Toll Free 1-877-525-2539

Outside US 1-818-487-4545

P.O. Box 15277

North Hollywood, CA 91615

FOUNDER

Jack Lemieux

PUBLISHER

Larry Lemieux

publisher@nutsvolts.com

ASSOCIATE PUBLISHER

Robin Lemieux

robin@nutsvolts.com

EDITOR

Bryan Bergeron

techedit-nutsvolts@yahoo.com

VP OF OPERATIONS

Vern Graner

vern@nutsvolts.com

ADVERTISING SALES

Garry Moore

adsales@nutsvolts.com

CONTRIBUTING EDITORS

Joe Pardue

Tim Brown

Jon McPhalen

Jeff Eckert

Paul Verhage

Chuck Hellebuyck

J.W. Koebel

Ron Newton

Dan Koellen

John Storms

CIRCULATION DEPARTMENT

subscribe@nutsvolts.com

SHOW COORDINATOR

Audrey Lemieux

WEB CONTENT

Michael Kaudze

website@nutsvolts.com

WEBSTORE MARKETING

Brian Kirkpatrick

sales@nutsvolts.com

WEBSTORE MANAGER

Sean Lemieux

ADMINISTRATIVE STAFF

Debbie Stauffacher

Re Gandara

Copyright © 2014 by T & L Publications, Inc.

All Rights Reserved

All advertising is subject to publisher's approval. We are not responsible for mistakes, misprints, or typographical errors. *Nuts & Volts Magazine* assumes no responsibility for the availability or condition of advertised items or for the honesty of the advertiser. The publisher makes no claims for the legality of any item advertised in *Nuts & Volts*. This is the sole responsibility of the advertiser. Advertisers and their agencies agree to indemnify and protect the publisher from any and all claims, action, or expense arising from advertising placed in *Nuts & Volts*. Please send all editorial correspondence, UPS, overnight mail, and artwork to: 430 Princeland Court, Corona, CA 92879.

Printed in the USA on SFI & FSC stock.



Well, first, expect the circuit to change. This means installing IC sockets for ICs instead of soldering them directly to a printed circuit board. It also means avoiding epoxy and other semi-permanent mounting schemes that could impede physical changes to the device later on. It also means that — given a choice — you should probably avoid soldering connections until you've made at least one improvement in the circuit. Breadboards and jumpers are a great way to explore the limits of a prototype. Wire wrapping provides a more stable/more compact circuit design than friction-fit jumpers — which is especially useful with mobile/portable prototypes.

Finally, how do you know you're "there?" That is, how do you know when your device is v1.0? That's where the art of prototyping is replaced by solid engineering principles. The engineering community has standards such as the common "mil spec" that define factors ranging from circuit operating conditions (such as ambient temperature range) to maximum electromagnetic interference (EMI) levels, and the ability to withstand G forces from rough handling or dropping. (Refer to the three-axis hockey-puck accelerometer data logger project we featured last month that will provide you with a device you can use to monitor how your packages are treated.)

Not surprisingly, many of the inexpensive devices on eBay are fancy looking prototypes. They're often not UL listed, and the "specifications" are at best taken with significant skepticism. For example, I've seen power supply specifications for unbelievably inexpensive units from China with specifications rivaling high-end power supplies from established suppliers such as HP, Kenwood, and others. Given the price, these inexpensive eBay units can be a bargain — especially if you recognize them as prototypes and are willing to invest some time in improving the circuitry so that it fits your needs. Perhaps replacing a flimsy heatsink with a more substantial one is all that's needed to create a workable power supply, for example.

Whatever changes are required, you'll certainly learn more about the underlying circuit than you would if you simply purchased a name brand commercial unit — and you'll save some money, as well. Enjoy prototyping. **NV**



EARN MORE MONEY

Get your dream job!

Be an FCC Licensed Wireless Technician!

Make up to \$100,000 a year and more with NO college degree

Get your "FCC Commercial License" with our proven Home-Study Course!

- No costly school. No classes to attend.
- Learn at home. Low cost!
- No experience needed!
- **MONEY BACK GUARANTEE:** You get your FCC License or money refunded!



Move to the front of the employment line in Radio-TV, Communications, Avionics, Radar, Maritime and more... even start your own business!

Call for **FREE** info kit: **800-877-8433** ext. 109

or, visit: **www.LicenseTraining.com**

COMMAND PRODUCTIONS • FCC License Training
Industrial Center, 480 Gate Five Rd., PO Box 3000, Sausalito, CA 94966-3000



ADVANCED TECHNOLOGY



■ 3D "dolls" made of gallium and indium alloy. The droplets adhere to each other but do not merge into a single droplet.

Antennas Made of What?

Given that the point of an antenna is to conduct electricity, it follows that metals with the highest conductivity make the best antennas. At the top of the list is silver, but several factors (cost, oxidation, structural strength) pretty much rule out its widespread use for such purposes. As you go down the conductivity list, you next hit copper which has many of the same problems. After that, you come to annealed copper, gold, and (we have a winner!) aluminum.

A few metals have rarely been considered as potential antenna materials, largely because they are liquids at or about room temperature and tend to be toxic or otherwise hazardous. For a year or so, however, researchers at North Carolina State University (ncsu.edu) have been assembling globs of liquid gallium (with some indium thrown in to keep it from freezing) via 3D printing that have potential transmit/receive applications.

Apparently, after you create an indium dot, wire, or other shape, its surface oxidizes and forms a skin over the fluid, producing something akin to a water balloon. "If you put aluminum into a rubber band, it will behave mechanically like aluminum," noted researcher Michael Dickey. "But if you put liquid metal into a rubber band, you have the metal conductivity — which you want — but you still have the properties of the rubber band. We are taking advantage of the fact that this metal forms this oxide layer in order to control its shape ... If you can change the shape, you can change the function."

This offers the possibility of new types of antennas for smartphones, navigation systems, Wi-Fi, and so on. It also could be adapted to wearable materials, soft robotics, and medical applications. Ongoing research is aimed at determining whether it is possible to modify the "skin" itself; if so, it may be possible to construct electrical components that are "adjustable," including smartphones with the ability to respond to changing environmental conditions.

The previously mentioned problems with silver are likely to apply here as well, as gallium (as of this writing) is selling for about \$30 per ounce vs. \$19 for silver. But there are always specialized applications. ▲

World's Fastest Electric

Electric motorcycles haven't exactly set the world on fire so far, but one sure burned up the Bonneville Salt Flats back in August. Builder and rider, Eva Håkansson managed to crush the former world's speed record for a battery-powered motorcycle by logging a top speed of 241.901 mph (389.219 km/h) and a two-way average of 240.726 (387.328 km/h) in her homebuilt vehicle, "KillaJoule." This beats the previous record by a remarkable 25 mph. It even breaks the record for any "sidecar motorcycle" (i.e., a three-wheeler in which the side wheel is not aligned with the rear wheel, unlike the standard motor tricycle), including those powered by internal combustion.

In fact, the last time an electric vehicle of any particular type beat the same type of internal combustion machine was 1899, when Camille Jenatzy ran an electric car — La Jamais Contente — up to a breathtaking 65 mph (105 km/h).

Håkansson and her husband, Bill Dubé describe themselves as backyard racers with high engineering skills, having manufactured about 80 percent of the vehicle in their garage on a shoestring budget. In fact, Eva is a PhD student at the NSF Center for High Voltage/Temperature Materials and Structures at the University of Denver, and Bill is a mechanical engineer and research scientist with NOAA, so "high engineering skills" may be a bit understated.

For those who are interested in specifics, the vehicle is driven by an EVO Electric AFM-240 motor (500 HP) and powered by an A123 Systems lithium nano-phosphate battery (375V, 10 kWh). It weighs about 1,540 lb (700 kg), including Eva. It measures 19 ft (5.6 m) long by 3 in (0.96 m) high, and has a 150 in (3.8 m) wheelbase.

And, no, you can't take it for a spin. ▲



■ Biker chick, Eva Håkansson and her record-setting "KillaJoule."

COMPUTERS and NETWORKING

Chromebook has 13.3 in Display

Many years ago, IBM was slow to get into the PC business because execs were convinced that the future of computing lay in the realm of a few huge supercomputers connected to many dumb terminals. This miscalculation allowed competitors to eventually force the company to divest its PC business to Lenovo. (It also was backhandedly instrumental in the creation of Microsoft, as IBM decided to get its OS from an outside vendor rather than spend the money to develop its own.)

Interestingly, because almost everyone now has a high bandwidth Internet connection, the concept to some extent is coming full circle in the form of Chromebooks: a category of cheap Internet-dependent laptops designed to be used primarily while connected to the Internet. Most applications and storage are "in the cloud," so as the Internet becomes "smarter," PCs can become dumber. Gartner projects that worldwide sales will nearly triple by 2017, reaching 14.2 million units.

One of the latest is Acer's Chromebook 13, billed as the industry's first to use an NVIDIA Tegra K1 4Plus-1™ quad-core ARM Cortex A15 CPU mobile processor; it's the company's first with a 13.3 inch display. Two display models are available: a full HD 1920 x 1080 resolution, and a 1366 x 768 version. The latter claims up to

■ Acer Chromebook offers a screen resolution up to 1920 x 1080.



13 hours of battery life, which drops to 11 hours with the HD screen.

Even though the unit is primarily for Internet use, the 3.3 lb (1.5 kg) machine does offer two USB 3.0 ports and an HDMI port for connection to a larger display. It also sports 720p audio/video recording with stereo speakers and microphones. Depending on the configuration, it will run you between \$279.99 and \$379.99. More details at us.acer.com. ▲



■ Western Digital's My Cloud® EX2 allows creation of your own private cloud.

Get Off Their Cloud

From the very beginning, one of the concerns about "cloud computing" has been the potential for criminals to hack into huge sensitive databases. In fact, it has happened so many times that it's almost old news. It is worth noting, however, that you can protect your personal data by simply creating your own personal clouds.

One available system — designed for "creative professionals, prosumers (i.e., amateurs who want professional-level equipment), and workgroups" — is the My Cloud® EX2, from Western Digital (www.wd.com). The unit is a two-bay network attached storage (NAS) device that allows you to protect your videos, photos, music, and other files using a choice of various drive management options. The key is the WWD My Cloud mobile and desktop app that provides access anytime, anywhere via iOS® and Android™ devices, as well as laptops and desktops.

For PC users, backups are implemented with the company's SmartWare Pro™ software, whereas Mac users can utilize Time Machine®. Standard drive capacities run from 4 TB to 8 TB. If that isn't enough, you can always move up to the EX4 and double up.

The entry level configuration (you have to add your own drives) is priced at \$199. For \$569, you get the loaded 8 TB version. It seems pretty reasonable compared to the potential costs involved with having your data stolen or destroyed. ▲

COMPUTERS and NETWORKING Continued

Take Your Own ECG

As with many tests and procedures performed at US medical facilities, it is tough to pin down how much you should expect to pay for an ECG (a.k.a., EKG). Various sources cite prices of up to \$3,000, but those appear to include not only the procedure itself but the analysis, as well. For the former, the numbers seem to cluster somewhere around \$200. So now, of course, there is an app for that.

Last August, AliveCor (www.alivecor.com) became the first company to receive FDA clearance for an algorithm that detects atrial fibrillation (AFib) — the most common form of cardiac arrhythmia. As of this writing, the company is still working to incorporate that algorithm into the app, but it should be available by the time you read this.

"The ability to automatically detect serious heart arrhythmia using mobile technology has the potential to save lives, reduce healthcare costs, and allow patients and their caregivers to make informed decisions about cardiac care," noted Euan Thomson, president and CEO.

Indeed, given that one in four adults over age 40 develop AFib, this can be an important extension of existing smartphone capabilities. To check yourself, you'll need an AliveCor heart monitor which is "intended for use by healthcare professionals, patients with known or suspected heart conditions, and health conscious individuals." It is, of course, recommended that you have a qualified healthcare professional verify the results. The monitor hardware is priced at (coincidentally?) \$199. It is compatible with all iPhone models and most Android devices. ▲



■ AliveCor's heart monitor can now perform ECGs via smartphone.

CIRCUITS and DEVICES

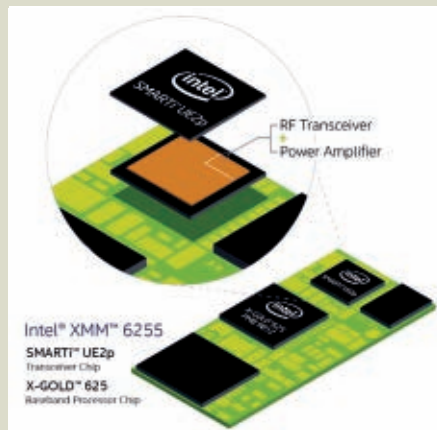
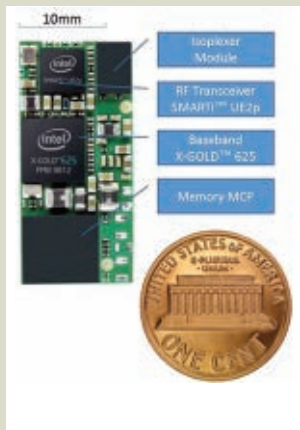
Intel Enters IoC

When the big dogs get involved in an emerging niche technology, that's a pretty good indication that it's about to become niche no more. It is therefore a likely bellwether that Intel (www.intel.com) recently announced the commercial launch of its XMM™ 6255 SMART™ UE2p Internet of Things (IoT) transceiver, billed

as the world's smallest 3G modem. According to the company, the design is "based on our unique new Intel® power transceiver technology — the industry's first design to combine transmit and receive functionality with a fully integrated power amplifier and power management, all on a single chip. This design approach reduces XMM 6255's component requirements, resulting in a smaller modem that helps manufacturers minimize their build of material costs. It also protects the radio from overheating, voltage peaks, and damage under tough usage conditions, which is important for safety monitors and other critical IoT devices."

Intel says the device provides a high level of low signal network coverage, which is important in places like parking garages and basements. It is specifically designed for use in very small devices such as smartwatches and sensors that may not have enough room for a normal sized 3G antenna. ▲

■ Intel's XMM 6255 IoT transceiver, said to be the world's smallest.

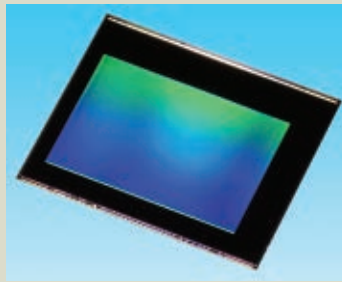


CIRCUITS and DEVICES Continued

20 Megapixel Sensor Introduced

If you have been disappointed in the quality and detail present in your selfies, hang onto your existing tablet or smartphone for a little while longer. Toshiba has launched a new 20 MP CMOS imaging sensor for both device types.

The T4KA7 incorporates a pixel size of only 1.12 μm , thereby allowing for higher resolution in 6 mm height camera modules. The sensor also offers an improved frame rate of 22 fps — nearly twice as fast as the company's previous 20 MP devices. The T4KA7 is presently available only in sampling quantities for about \$19. Production will be ramped up to 500,000 pieces per month as demanded by market forces. ▲



■ Coming to high end smartphones and tablets: Toshiba's 20 MP CMOS sensor.

INDUSTRY and the PROFESSION

Preserving the CRT

The migration to flat screen displays has nearly wiped CRTs from the face of the earth. The last CRT rebuilding operation in the USA — Hawk-Eye Picture Tube Mfg. — closed down in 2010 after 52 years in the business, and the last one in the UK — RACS — shut down last year. For most people, this is not a bad thing. CRTs are heavy, bulky, and suck down a lot of juice. Flat displays are just the opposite, plus they're cheap these days. However, some people are into classic TV sets and enjoy the retro effect. Others might want to keep the antique oscilloscope working.

If you fall into any such category, take heart, because the Early Television Museum (www.earlytelevision.org) is planning an upcoming launch of its CRT Rebuilding Project. Scott Avitt, who in 1976 took over Hawk-Eye from his father, has offered to donate his equipment to the museum and teach the secrets of rebuilding tubes. The rebuild facility is to be located in the museum's facility in Hilliard, OH (northwest of Columbus).

The museum itself is supported by tax-free donations, so feel free to lend support while earning a little tax dodge. The museum is also seeking equipment donations — particularly American and British scanning disk sets, American and British pre-1945 electronic sets, unusual 1946 to 1949 sets, and experimental and color 15 inch sets. This could be a good time to clear out some space in your workshop.

If you happen to be passing through Hilliard, stop in for a tour; they're open Saturdays from 10:00 to 6:00, Sundays from noon to 5:00, and other times by appointment. Providentially, there's a White Castle just four miles down the road, so you can kill two birds with one stone. **NV**

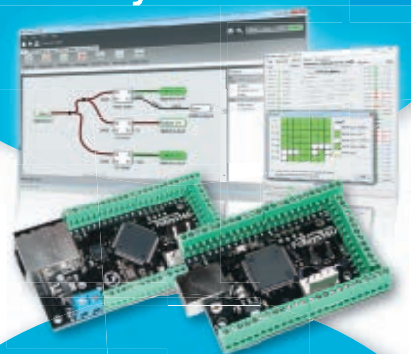


■ RCA 630-TS, the first mass-produced TV set (1946-47).

Imagine this...

- a small and inexpensive
- USB or ethernet device
- with a rich set of features
- including PLC functionality and up to 8-axis stepper motor controller
- all accessible via free .NET, ActiveX or C++ library
- cross-platform
- configurable with free software

PoKeys



Or this...

- all in one: Oscilloscope, Data Recorder, Logic analyzer, Analog and digital signal generator
- smallest USB 2.0 portable 1MS/s oscilloscope
- data acquisition of analog and digital signals
- data recording
- export to CSV, XLS, PDF and HTML
- simple usage of advanced features
- examples for C++, VB, Delphi and LabView
- free software and updates

PoScope Mega1+



Polabs
Visit us at
www.poscope.com



In this column, Tim answers questions about all aspects of electronics, including computer hardware, software, circuits, electronic theory, troubleshooting, and anything else of interest to the hobbyist. Feel free to participate with your questions, comments, or suggestions. Send all questions and comments to: Q&A@nutsvolts.com.

- **Battery Monitor Circuit**
- **NiMHs or NiCADs**
- **Audio Synthesizer IC With 8 Ohm Speaker**

Post comments on this article at www.nutsvolts.com/index.php?/magazine/article/december2014_QA.

Need Battery Monitor Circuit

Q I'm trying to design circuitry that shows a low battery (once the voltage goes below a certain limit, e.g., 3.1V or 3.2V) by flashing an LED (0.1 sec every two to four sec). I'm using an Li-Ion battery 14500 series and need to monitor the device before the battery goes dead. I have to say there are many designs for this purpose, but I'm seeking a design that has the least impact on power consumption.

—Sam Jannati

A Lithium-Ion batteries (a.k.a., Li-Ion) use lithium-ions moving from the negative electrode (cathode) to the positive electrode (anode) to produce a flow of current in an electrical circuit. To recharge the Li-Ion battery, you reverse the direction of current flow. Li-Ion batteries do not need the "conditioning" required by their Nickel-Cadmium (NiCads) or Nickel-Metal Hydride (NiMH) cousins. However, Li-Ion batteries are VERY sensitive to operating outside the range of three to 4.2 volts (overheating, fire, explosions). Manufacturer specified chargers have the circuitry to avoid overcharge, and battery packs have circuitry to avoid operating below the three volt shutoff voltage. CAUTION:

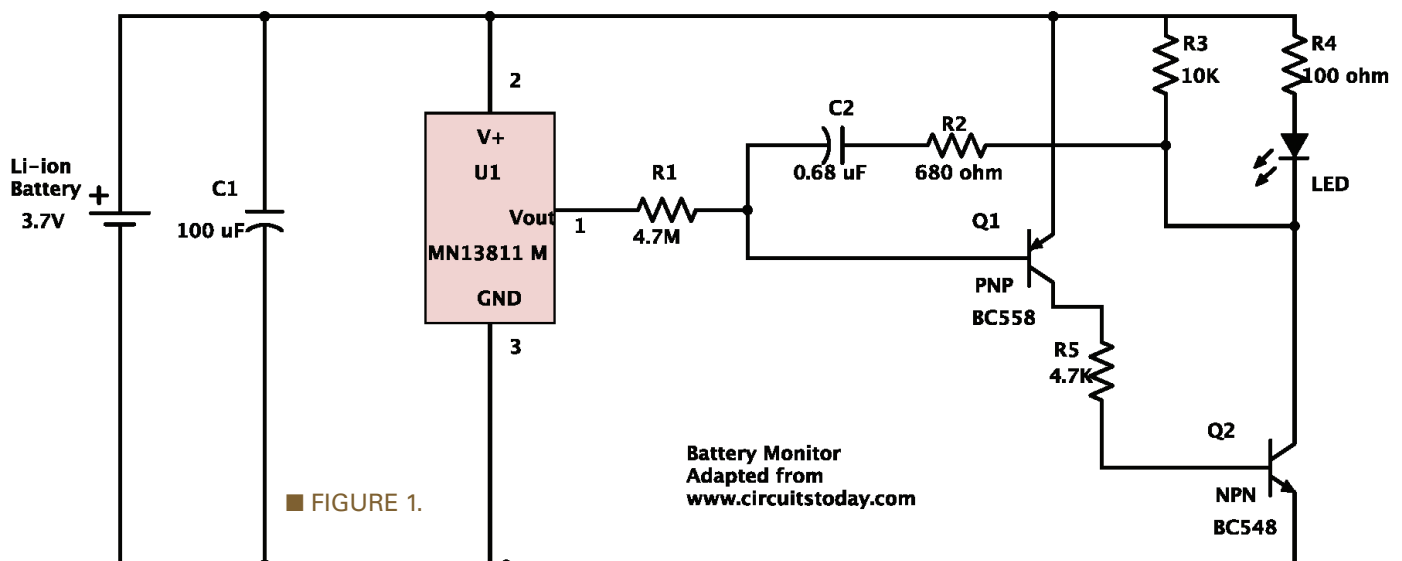
Always follow manufacturer's recommendations for charging and operating batteries — laptops, cell phones, and even a factory have burned down due to Li-Ion batteries.

A simple monitor circuit could use an operational amplifier (op-amp) to compare the Li-Ion battery's voltage to a zener diode reference voltage to turn on an LED for low battery voltage. Your requirement for a flashing LED, however, dictates the use of a voltage detection IC such as the MN13811-M which has a 3.2V voltage threshold.

Figure 1 shows a circuit schematic diagram for the low voltage monitor adapted from circuitstoday.com. A special thanks to Digi-Key for their free online schematic drawing software, Schemelt (www.digkey.com/schemeit#).

NiMH Batteries Won't Replace NiCads

Q I have an outdoor solar lantern. The original circuit died, and I have replaced it pretty much exactly as it was. I noted that candles don't actually flicker off and on, but the flame moves. So, I have two LEDs controlled by a PIC. I have it run for



six hours and then shut down. The circuit is in **Figure 2**.

My problem is that NiCad batteries work fine, but are getting hard to find. So far, I can get them from Mouser, but I wonder how long that will last. NiMH batteries work for a few days, then quit. It may be that they run until the original charge is used up and never charge at all.

In the sun, the input to a NiCad pair is 3.62 volts after two-thirds of a day's charging. It would not be practical to add another solar panel if I need more voltage to charge NiMH batteries. The diode has the lowest forward voltage drop I know of.

Is there anything else I can do to accommodate NiMH batteries? I would like to know what is going on. I thought that NiMH batteries were different somewhat from NiCads, but pretty close — at least close enough for a circuit as simple as this one.

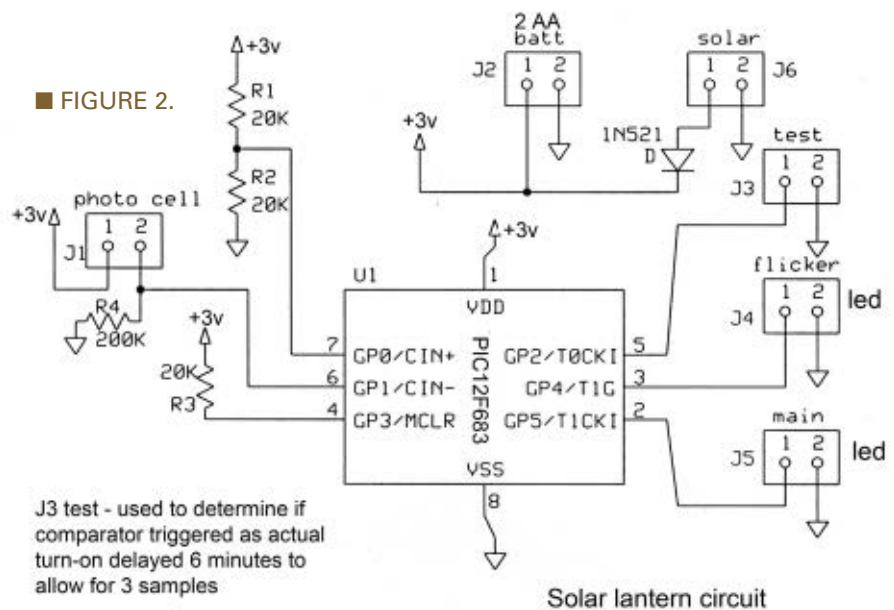
— Alan May (still using a Heathkit 1500 amp)
Houston, TX

A I am glad to see someone else using older but proven equipment (I still use a Simpson meter — for those who don't know what this is, Google it). At the electronics program at the community college where I taught, we had several Heath power supplies dating to the late '70s and they kicked the butts of the newer, slicker digital supplies.

As to your question, NiCad (Nickel-Cadmium) and NiMH (Nickel-Metal Hydride) batteries are very different "animals." Besides the differences in chemistry (for those non-chemistry readers, I won't bore you with this discussion, but for those interested in the chemistry check the Internet because it would take more space to explain that I am allowed here), the batteries have different charge and discharge characteristics. NiMH batteries self-discharge at twice the rate of NiCads of the same ratings. (In plain speak, if the NiCad lasts six weeks on the shelf, the NiMH in an unpowered device will only last three weeks before it needs a recharge.)

NiMH charge rates must be lower than that of NiCads to avoid damaging the NiMH battery. NiMH batteries will last longer, but have a lower maximum discharge rate than NiCads. NiMH batteries are very sensitive to over-charging, which I suspect your charger is doing. Charging NiMH batteries requires a circuit that detects state-of-charge so the battery is not over-charged, and the solar flashlight may not have enough room to install this additional circuitry.

A low value resistor in series between the diode and



battery may reduce the charge rate sufficiently, but you will have to experiment to find a suitable value. As to NiCAD batteries being hard to find, that's a new one to

Start learning electronics with an all-in-one lab!

- * Audio Range Function Generator
- * Audio Range Oscilloscope
- * Spectrogram Function
- * Three Power Supplies
- * Large Breadboard
- * Free Ebook

Explore the world of analog electronics with our all-in-one learning system. The free ebook will teach the basics with labs you can build and verify using our Analog Bakerboard Educational System!



LumiDax Electronics LLC

www.lumidax.com

Phone: 541-510-9264

me since the Internet has many vendors that supply these types of batteries (this may be true if your device has

**WORLD'S MOST VERSATILE
CIRCUIT
BOARD HOLDERS**

Model 324 Our Circuit Board Holders add versatility & precision to your DIY electronics project. Solder, assemble & organize with ease.

VISIT US ON

MONTHLY CONTEST
 Visit us on Facebook® to post a photo of your creative PanaVise project for a chance to win a PanaVise prize package.

PANA VISE®
 Innovative Holding Solutions

7540 Colbert Drive • Reno • Nevada 89511 | (800) 759-7335 | www.PanaVise.com

OEM batteries instead of a standard battery pack, but AA batteries are a standard).

Not knowing the programming you have in the PIC, I suspect if you also changed the PIC, there may be a faulty line or more of code which controls the "flicker" function. The function of the diode is to prevent discharging the battery into the solar cell when the solar cell voltage drops below that of the battery (a solar cell or photovoltaic cell is a P-N junction which generates a current when light falls on the junction, which is connected as reverse biased).

Using an Audio Synthesizer IC With an 8 Ohm Speaker

Q I'm playing around with an ancient AY-3-1350 tune synthesizer IC and am having trouble with the audio output portion. From the looks of the schematic in the datasheet, the speaker is 80 ohms! I have never seen or even heard of an 80 ohm speaker. Where would I find such a dinosaur? Or, in lieu of that, it looks like the IC is modulating both sides of the audio amplifier circuit. Is there a more modern version of the audio amp portion that I could use that drives a standard 8 ohm speaker?

— Derek Tombrello

A Derek Tombrello is the honcho at **RobotsAndComputers.com** (a good place to look for info on all things robot and computer — well, duh!!!) and has contributed articles to *N&V*, so in answering this question I feel like the guy killed in the Johnstown flood telling Noah about issues with water.

The AY-3-1350 is an application specific IC (ASIC) chip manufactured by General Instruments for generating tunes and chimes. By selecting the letter and number "address" lines, you can play 25 tunes and three chimes, with the capabilities to play user-defined tunes to boot. Look at the datasheet at sources such as www.elenota.pl/datasheet-pdf/67443/others/AY-3-1350 since it is too extensive to print here. The datasheet gives a great setup for playing tunes, but has an 80 ohm speaker. The number says 80 ohms, but measured with your ohmmeter you will get a much smaller reading. The speaker "ohm" spec is impedance (a combination of resistance, capacitive reactance, and inductive reactance of which the latter two are frequency dependent) which is measured at

ARLCD 3.5-inch Arduino GPU Combo Just \$99.00



Product Specifications: 3.5" color TFT LCD • 320 x 240 Resolution • 65k colors • Touchscreen • Powerful 16-bit microcontroller GPU • 4MB flash memory for storing fonts, bitmaps & macros • USB 2.0 • Overall outline dimensions: 3 x 3 x .9 inches • 6-9V operating voltage • Extremely low power - draws less than 200mA • ARLCD Arduino GUI Library • Arduino UNO R3 Compatible



400 Hz. Standard speaker impedances are 4, 8, 16, 45, and 80 ohms, depending on the application.

Why should I worry about speaker impedance and not just hook up an 8 ohm speaker instead of the 80 ohm speaker shown in the datasheet schematic? (THEORY ALERT: Theory — or why — is equally as important as knowing how.) The short answer is the Maximum Power Transfer Theorem which, in essence, says: "The maximum amount of power is transferred from the source (amplifier output) to the load (speaker) when the input impedance of the load matches the output impedance of the source."

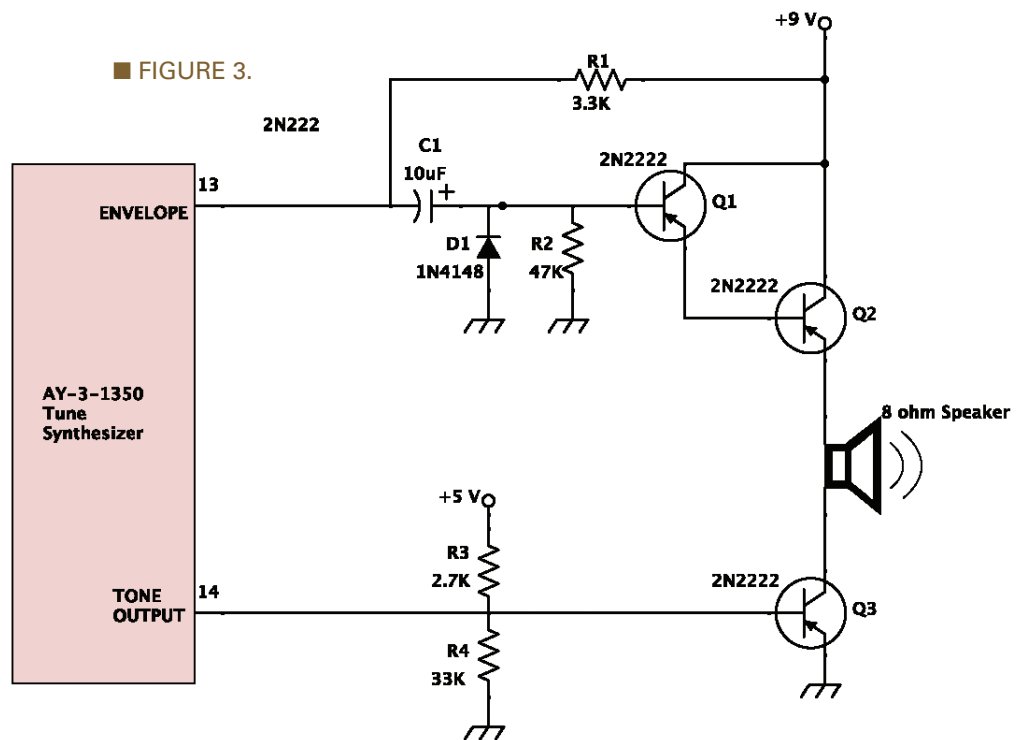
Install an 8 ohm speaker in place of an 80 ohm speaker of the same power ratings and you will get less than maximum power to the speaker, which translates into less volume. Plus, the source's output circuits may experience overheating because the excess power that should be transferred to the speaker has to be dissipated in the output circuitry (I had a student bring in a stereo amp in which one of the channels did not work. After opening the case, I saw the problem and told the student, "Where you see those three "legs" ... there should be a power transistor like the one on the other side.").

The Law of Conservation of Energy says that energy and power cannot be destroyed, so the power from the amplifier output has to go somewhere — either load or source. Also, it does not matter which way the imbalance goes — high output Z to low input Z, or vice versa — it is the mismatch that counts.

I found a circuit for the AY-3-1350 using an 8 ohm speaker at http://matthieu.benoit.free.fr/ay-3-1350_General_Instruments_resources_page.htm for which I have re-sketched the output circuit in **Figure 3**. A special thanks to Digi-Key again for their free online schematic drawing software.

CAVEAT: You may have to experiment with the resistor values for optimum performance, and don't expect to drive high wattage speakers — the 2N2222 is rated at 0.5 watts maximum collector power dissipation. Check out Matthieu Benoit's website for some cool ways to operate the AY-3-1350 synthesizer. I did not include the whole schematic here, just the output which was the subject of the question. **NV**

■ **FIGURE 3.**



2 Layers

\$3
ea.

- FULL DRC CHECK
- CAD/CAM REVIEW
- SOLDER MASK ON BOTH SIDES

www.PCB4u.com
(408) 748-9600

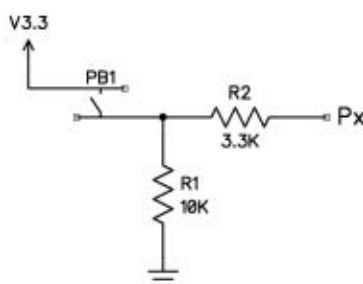
PRINTED CIRCUIT BOARDS

Put a Button On It

New actors are often encouraged by casting directors to "put a button on it" — this means to end the piece with something simple yet memorable. Well, then, let's put a button on 2014 by talking about ... buttons! In our high tech world of touchscreens and voice-activated everything, simple buttons are often abused both physically (which I can do nothing about) and in code (where I can help). The simple fact of the matter is that most electronic devices have buttons of some sort, hence we should learn to use them properly. Whether your project uses just one button or a bunch of them, I'm going to show you some tips and tricks that can be incorporated into your Propeller applications.

I think it's Nature's cruel trick that the smallest things tend to bite the hardest — I learned this lesson well growing up in the desert catching lizards and snakes. The same holds true with buttons and switches. If we don't treat them well, we may end up with a nasty surprise. I'll give you a real world example.

One of the entertainment venues that I work in is the Halloween industry. Most modern Halloween attractions use a wide range of electronic controls. An EFX-TEK customer called me from New York in an absolute panic — every time a walkie-talkie was keyed on, random props around his haunt would start operating. This was not the kind of nightmare he was looking for in his attraction.



■ FIGURE 1. Button circuit.

Button Basics

In **Figure 1**, you see a typical button input circuit. This is configured for active-high input to the pin (you can reverse the 3.3V and ground connections to make it active-low). When

the button is open, the pin will read 0; when the button is pressed, the pin will read 1.

Some will ask about resistor R2. This resistor limits the current into the pin if there is a programming error or if the input voltage is 5V (which is frequently present in Propeller projects), plus it protects the pin (as well as one can) from static discharge we humans tend to barrage electronic devices with.

Simple, right? Of course — but there's more to it.

We tend to think of a short button press creating an input on the pin that looks like **Figure 2**. If we put a 'scope on the input, what we'd see looks more like **Figure 3**. The ragged on and off switching at the beginning of the input is called contact bounce.

This is, in fact, caused by the mechanical contacts of the button bouncing as it closes.

Some will say, "So, what?" Okay, let's get back to my friend in New York. His particular situation created a perfect storm of problems with programs using less-than-industrial techniques. His props used very long wires to connect physical triggers (buttons and switches) to the controllers. Long wires act like antennas. A radio that's transmitting generates RF that can be picked up by these long trigger wires.

The voltage induced across the trigger wire created noise on the trigger pin. Inappropriate trigger code in the controller allowed this noise to create a false positive, and his props went crazy. Thankfully, the solution was simple — so simple it is often overlooked.

Before we get to the solution, let's have a look at the problem. Here is an atrocious (yes, atrocious) example of button input code:

```
pub bad_button
    if (ina[0] == 1)
        return true
    else
        return false
```

The only redeeming quality of this method is that it can be re-used. Where it goes off the rails is in two places: 1) The input pin is hard-wired as a "magic number;" and 2) The scan of the input pin is so quick that it can allow a false positive if noise is present on the pin when this method is called.



■ FIGURE 2. Clean input.



■ FIGURE 3. Bouncy input.

Post comments on this article and find any associated files and/or downloads at www.nutsvolts.com/index.php?/magazine/article/december2014_SpinZone.

If you're writing code like this, please stop. Right now. Using "magic numbers" is just lazy, and you're only pouring oil in your own path. Stop it!

I know, I know, you're just writing a "quick and dirty" program. Bollocks. These kinds of programs are rarely quick, and they're always dirty. Please trust me that it takes no more time to code properly than to write quick and dirty — and properly written code can save you hours of debugging time. Replace your magic numbers for pins with named constants.

That said, simply using a named constant for the pin does not save the above code. What we want to do is scan a button and check for a valid input — not just transient noise on the pin. We can do this with a process called debouncing.

A very simplistic debouncing process — which I don't like — goes like this:

```
" Check input
" Wait (e.g., 25ms)
" Check input again
```

If the first and second checks match, the input is considered valid. My own debouncing method is somewhat more involved, yet very easy to implement. Instead, there are two checks (at the beginning and end of the debounce period). I divide the debounce period into small units (1 ms), re-checking the pin(s) at the end of each unit. This forces the input to be re-checked several times during the debounce period.

Here's an example of a good button input method that uses my debouncing style:

```
pub check_button(pin, ms) | btn, t

  btn := 1

  t := cnt
  repeat ms
    waitcnt(t += MS_001)
    btn &= ina[pin]

  return btn
```

Note that this method expects two parameters: the pin to scan, and the debounce period (in milliseconds) — these parameters give us great flexibility.

As we've learned our lesson about taking things for

BOM

ITEM	DESCRIPTION	SOURCE/PART #
PB1	12 mm, N.O.	Mouser 101-0621-EV
PB1 Cap	Black	Mouser 101-0210-EV
R1	10K, 1/8W	Mouser 299-10K-RC
R2	3.3K, 1/8W	Mouser 299-3.3K-RC

granted, the first thing this method does is set the target pin to input mode. The second step may seem odd, but will make sense in a moment. We set the working result (in *btn*) to 1; this enables the result for the scan code that follows.

The loop is set up with a timer variable, *t*, that is initialized to the value in the system counter. The **repeat** loop will be run the number of iterations passed in *ms*; hence, the delay at the top of the loop is set for one millisecond (using **waitcnt**).

After the delay, we scan the input, then AND (&) it into the result value. This is the trick: If the switch bounces open, the input will be zero. If we AND anything with zero, we get zero — this will keep the result at zero for the rest of the scan. What this means is that the button must be pressed and stay pressed during the entire debounce period. The result will be one if the button was properly debounced; zero if not.

As Spin can treat any non-zero value as true, we can use the **check_button()** method like this:

```
if (check_button(TRIGGER, 25))
  play_show
```

Some of you may be wondering about active-high versus active-low inputs. My preference is to use active-high as it's easier for my clients (who don't code as much as I do) to follow. Still, many use active-low circuits, and it's easy to update our method to allow either input state:

```
pub check_button(pin, state, ms) | btn, t

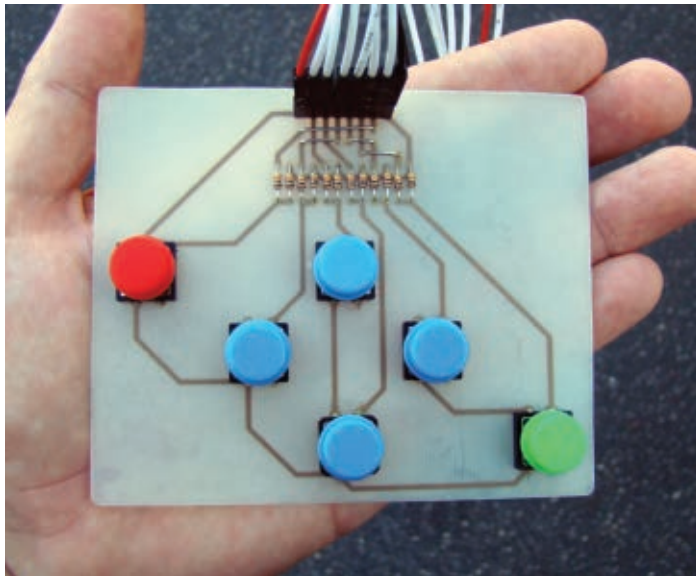
  dira[pin] := 0

  btn := 1

  t := cnt
  repeat ms
    waitcnt(t += MS_001)
    if (state)
      btn &= ina[pin]
    else
      btn &= !ina[pin]

  return btn
```

Note that we've added an input state parameter (1 for active-high; 0 for active-low), and the scan code will invert (!) the button input if the state has been set to active-low. Easy peasy.



■ FIGURE 4. Nav button board.

Multiple Inputs

Most of my applications use multiple buttons, and dealing with them is no more difficult than with a singleton. In **Figure 4**, there is a simple navigation button board I knocked together on a home-etched PCB (printed circuit board) using the PulsarProFX PCB Fab-In-A-Box system. This uses six copies of the circuit in **Figure 1**. I could have wired it on a perfboard, but I have next to no patience for point-to-point wiring. The PulsarProFX system makes etching single-sided PCBs very simple, and I feel I can etch a PCB more quickly than I can hand-wire one.

Let's look at a debounce method for a contiguous group of buttons:

```
pub check_buttons(msb, lsb, state, ms) |
  btns, t

  dira[msb..lsb] := 0

  btns := -1 >> (31-|| (msb-lsb))

  t := cnt
  repeat ms
    waitcnt(t += MS_001)
    if (state)
      btns &= ina[msb..lsb]
    else
      btns &= !ina[msb..lsb]

  return btns
```

As you can see, there's very little difference monitoring multiple buttons versus one. With this method, we will pass the pin group boundaries in *msb* and *lsb*; I find it cleanest to deal with multiple buttons in a contiguous group.

Enabling the inputs that we want to scan might look a tad gnarly to newcomers, but it's really not so bad. This is the line:

```
btns := -1 >> (31-|| (msb-lsb))
```

What this does is fill *btns* with as many 1s as we have input pins. If our inputs are 5..0, *btns* would be initialized to %111111.

The binary representation of -1 is all ones (32 of them!). What we want to do is shift this to the right to pad the left side of the value with zeroes. The span of the shift depends on the number of input pins we're using. The number of input pins (less one) is calculated by taking the absolute value (||) of the *msb* pin minus the *lsb* pin.

We use the absolute operator so that the *msb* pin number can be smaller than the *lsb* pin number (more on this in a moment). Since the pin count is one short of what we have, we subtract from 31 instead of 32. After shifting -1 the correct number of bits, we have a mask that matches the number of pins we're scanning.

The rest of the code is identical to the single-pin version; the exception being we are using a range of pins instead of one. That said, we can simplify listings by using **check_buttons()** for everything; it will work with one to 32 pins.

Before we move on, let me comment on the use of *msb* and *lsb* in the routines we've just created, and do it in practical terms with my nav button board. I have six buttons that I define with these masks:

```
con
  MASK_UP   = %000001
  MASK_DN   = %000010
  MASK_LF   = %000100
  MASK_RT   = %001000
  MASK_OK   = %010000
  MASK_ESC  = %100000
```

I have used this simple paradigm in a number of personal and professional projects; the latter group includes a sprinkler controller, a camera platform controller, and even a large-scale HVAC system controller.

Note how the masks indicate that the UP button is in the LSB position of the group. What happens, though, when I get to the PCB layout and it would just be easier to have connections reversed? Yes, I could go in and change my masks. With Spin, though, I don't have to. By passing the LSB and MSB *connections* to the routine, the result will always match my masks.

Let's say I plan a circuit to use P0 through P5, with P0 being my UP (LSB) button. In this case, the LSB to the call would be zero and the MSB would be five (as named constants, of course). However, I get to the PCB layout and find the board would be cleaner (i.e., fewer vias) if I flopped the inputs. No problem — by flopping the LSB

(to 5) and the MSB (to 0) in the call to **check_buttons()**, I will still get results that match my masks (i.e., bit 0 of the result will be the UP button).

Spin takes care of this for us when specifying input and output ranges. It's probably clearer now why we used the absolute operator (|) when creating the pin's mask; this allows the *msb* pin to be a lower number than the *lsb* pin.

Background Button Scanning

One of the advantages we have with the multi-core Propeller is the ability to create background processes that don't interfere with our main program's timing. This allows our applications to look at a simple variable as if it were a group of pins — the difference is that the bits in that variable represent inputs that have been fully debounced.

I've created a simple object called **jm_buttons.spin** that lets me set up a number of contiguous inputs to be debounced, the state (active-high or active-low), and the debounce period. The **start()** method is set up very much like our **check_buttons()** parameters:

```
buttons.start(PB_ESC, PB_UP, 1, 25)
```

Once it's running, we can make a simple call like this:

```
btnstatus := buttons.read
```

The **read()** method returns the entire group. If we only want to know the state of a specific input, we can pass its offset in the group to the **rd_bit()** method. For example, in my navigation panel, the UP button is at offset 0:

```
if (buttons.rd_bit(BTN_UP))
    state += 1
    pause(50)
```

Real World Solutions

While code re-use should be encouraged to the greatest possible degree, the real world will at times dictate specific solutions. For example, the EFX-TEK HC-8+ controller has four active-low configuration inputs that are controlled by a DIP switch, and a header allows the customer to extend these inputs to external buttons and switches. Since the inputs go directly to the Propeller, they should be debounced. No problem.

The HC-8+ also has a bi-color LED. It's very simple to make it off, red, or green, but making it yellow requires switching between red and green at a specific ratio. While we would normally use a bi-color LED object, in this instance I decided to write project-specific code that solves both problems (debouncing and LED color control) in a single cog.

Even in Spin, we can accomplish quite a lot in one millisecond. For my bi-color LED control, I even go another step and provide two phases, with each phase being assigned a color and duration (in milliseconds). If the colors in each phase match, the LED is solid. If the colors are different, it blinks — the rate of blinking is controlled by the time settings in each phase.

Let's break down the method that handles LED control and inputs debouncing:

```
pri run_rgy | t, {
} rgphase, phasetimer, rgtimer, {
} dbwork, dbtimer

outa[R_LED..G_LED] := %00
dira[R_LED..G_LED] := %11

rgphase := 0
phasetimer := 0
rgtimer := 0

dbwork := %1111
dbtimer := 0
```

This part is very easy; it's the setup which includes making the LED pins outputs in this cog (remember, each cog has its own output control register) and initializing the LED and debounce variables:

```
t := cnt
repeat
    waitcnt(t += MS_001)
```

No worries here, either. This is at the top of the loop. We initialize the loop timer control variable (t), set up the repeat loop, then insert a 1 ms **waitcnt**. It really doesn't matter where the **waitcnt** falls, but for longish loops like this I tend to stick it at the top:

```
if (++phasetimer >= rgytime[rgphase])
    phasetimer := 0
    rgphase := 1 - rgphase

if (++rgtimer == 20)
    rgtimer := 0

case rgycolor[rgphase]
    OFF:
        outa[R_LED..G_LED] := %00

    GRN:
        outa[R_LED..G_LED] := %01

    RED:
        if (rgtimer < 4)
            outa[R_LED..G_LED] := %10
        else
            outa[R_LED..G_LED] := %00

    YEL:
        if (rgtimer == 19)
            outa[R_LED..G_LED] := %10
        else
            outa[R_LED..G_LED] := %01
```

So, now we get into it. The first step in LED processing

is incrementing the phase timer and then comparing it to the setting for the current phase. When these values match, the phase timer is reset, and the phase is flipped (0 to 1, 1 to 0). The phase is used as an index into the LED color and timing arrays.

A second timer is used for LED control. A case structure handles setting the base color, and the color timer (*rgtimer*) is used when the setting is red or yellow. The reason for controlling the red chip is that it's so much brighter than the green. By running the red only four cycles out of 20, the red chip brightness better matches the green. This is best seen when the LED is set to a slow blink between red and green.

For yellow, I use a ratio of one red cycle to 19 green cycles. Surprising, isn't it? I'll bet you thought it would be a 50/50 mix. Again, the brightness of the red chip in the LED is significantly greater than the green, hence the lopsided mix.

The final section of the loop scans the configuration bits and updates a user variable after the preset debounce period. This should look familiar:

```
dbwork &= !ina[OPT_BR..OPT_SM]

if (++dbtimer == DBMS)
  optbits := dbwork
  dbwork := %1111
  dbtimer := 0
```

When the debounce timer expires, we update *optbits* (which is global to the program), then reset for the next cycle by reinitializing the inputs and resetting the timer.

Again, try to keep your code as generic as possible, but don't handcuff yourself with unnecessary flexibility; when you have a fixed application, fix the code to it.

Smart Button

Let me share one more trick before we all head off to holiday parties. In early July, I went to a cosplay event to visit my friend, Aubriana (www.themadmasher.com). While there, she introduced me to Jinyo, one of the cast members of the SyFy Channel show, *Heroes of Cosplay*. When I told him what I do, he asked if I might be able to help him with a project for Comic-Con International in San Diego, CA. Sure!

Jinyo and his fiancée, Victoria, created a beautiful Cylon-themed dress for a Comic-Con fashion show. Built into the dress is a long strip of WS2812 LEDs that are controlled with a Propeller Mini. I built a small carrier PCB with sockets for the Mini (using the PulsarProFX system); this also provides connectors for the battery pack and LEDs.

While working in his shop, Jinyo asked if we could add a button to allow the model to control the LEDs. Sure! What he ultimately wanted was the ability to cycle

through various animation states, and to be able to reset to the beginning (OFF) at any time. Mind you, we only have one button (which Jinyo cleverly built into the waist of the dress).

The solution I used was to create a background cog that would scan and debounce the button, and measure timing so the program could determine a "normal" versus "reset" press.

Let's have a look:

```
pri monitor_button(pin) | scan, t, dbtimer
  dira[pin] := 0

  scan := %00
  dbtimer := 0

  t := cnt
  repeat
    waitcnt(t += MS_001)

    scan := ((scan << 1) & %11) | ina[0]

    case scan
      %00, %01:
        dbtimer := 0

      %11:
        dbtimer += 1

      %10:
        if ((dbtimer == DB_MIN) and (dbtimer
          =< DB_MAX))
          if (state <> PARTY_STATE)
            state := ++state // N_STATES

        elseif (dbtimer == DB_RESET)
          if (state <> 0)
            state := 0
          else
            state := PARTY_STATE

    else
      dbtimer := 0
```

There are a number of ways to measure the valid state of an input. The approach here is to run the scan in a one millisecond loop using a case structure to update the debounce timer — based on the button's current and previous state.

The variable called *scan* holds those states. Each time through the loop, we update *scan* with this line:

```
scan := ((scan << 1) & %11) | ina[0]
```

This starts by shifting *scan* one bit to the left, and moving the previous button state into *scan.bit1*. We do a bit of clean-up using AND (&) which limits *scan* to two bits. Finally, the state of the active-high button is moved into *scan.bit0*.

Here's how it works. If the value in *scan* is %00, the button is not pressed, nor was it pressed during the last cycle. If the value is %01, we have detected a new button press. In either case, the debounce timer (*dbtimer*) is

reset. When the value in *scan* is %11, the button is being held down, hence the timer is incremented. Finally, when the value in *scan* is %10, we have detected the falling edge of the button input. This could, of course, be contact bounce.

You can see that most of the work is done when *scan* is %10 (end of input). The first step is to compare the timer to a normal press (which we set from 50 ms to 500 ms). If this is the case and the program is not in "party" mode (more on that in a moment), the animation state control variable is incremented.

If a long button press is detected, the animation state is checked before acting. If we're in animation mode (for the runway), we reset the state to off. This was important to allow Jinyo to change to a fresh battery pack before his model went out.

After changing to fresh batteries, he could press the button to verify everything is working. With that test done, a long button press would reset the animation to off.

Those of you who have been to a Comic-Con understand that they are very social events, and Jinyo envisioned his model wearing the dress to a party after the fashion show. For this, he requested a "party" mode that was a standard Larsen scan of the LEDs, but at a significantly reduced brightness. WS2812s can be very bright, and in the dim environment of a cocktail party, he didn't want the dress blinding other guests!

Of course, for the normal animation mode on the runway, we used full brightness. (By the way, the dress was a huge hit!)

Putting a Button on 2014

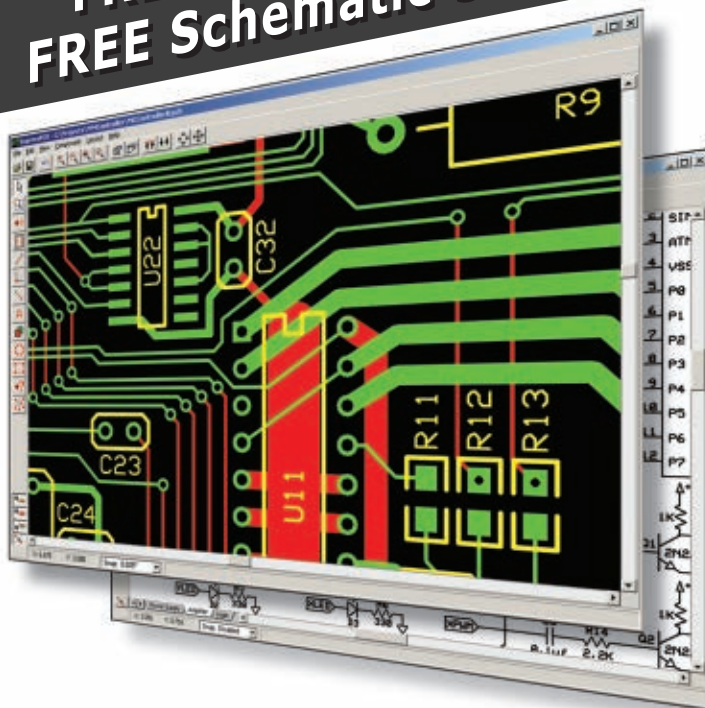
Well, we've done it! We've completed another year ... well done, us! I'm sure that more than a few of you started reading this column thinking, "Really? How much is there to say about button inputs?" Quite a lot, actually, and I hope you found this discussion and

the tricks I shared useful.

From my family to yours, I wish you the best this holiday season. Happy Hanukkah! Merry Christmas! Happy New Year!

Until next time, keep spinning and winning with the Propeller! **NV**

\$51^{For 3} PCBs
FREE Layout Software!
FREE Schematic Software!



- 01 DOWNLOAD our free CAD software
- 02 DESIGN your two or four layer PC board
- 03 SEND us your design with just a click
- 04 RECEIVE top quality boards in just days

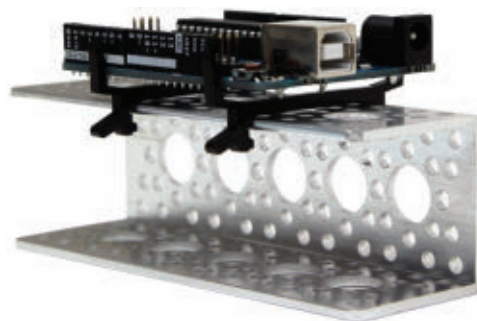
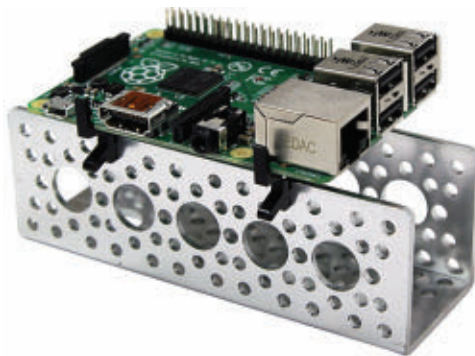
expresspcb.com

NEW PRODUCTS

- HARDWARE
- SOFTWARE
- GADGETS
- TOOLS

CHANNEL SNAP MOUNTS

New Actobotics channel snap mounts from ServoCity make it simple to attach electronics and other components to aluminum channel. These new snap mounts provide a quick and simple way to mount an Arduino (UNO or Mega) or Raspberry Pi microcontroller to an Actobotics structure. Just snap the mounts onto the aluminum channel and press-fit a microcontroller board onto the top of the mounts. The versatile snap-on components can attach to channel in multiple configurations, providing lots of mounting possibilities. Made from 1/8" thick Delrin, the mounts are both lightweight and very sturdy; pricing is \$1.89 per set.



DONGBU SERVO GEARS

In addition to their patented line of Hitec and Futaba servo gears, ServoCity now has a way to drive gears using Dongbu servos. Dongbu Herkulex robot servos are high quality and innovative servo motors with a 25 tooth spline. ServoCity's 32 pitch gears attach directly to the DRS-0401 and DRS-0601 Herkulex servos. All gears are broached directly into the metal to ensure a snug fit that will not slip. The metal construction provides longevity and durability. They are currently available in a 16-tooth version (#615382), with more sizes to come soon. Pricing is \$14.99/each.



For more information, contact:
ServoCity
Web: www.servocity.com



PBP3

PICBASIC PRO™ Compiler
BASIC compiler for Microchip PIC® microcontrollers

FREE
Student - ~~\$49.95~~
Silver Edition - \$119.95
Gold Edition - \$269.95

Download **FREE** trial now

Simple, Reliable, Affordable

WWW.PBP3.COM

PICBASIC and PICBASIC PRO are trademarks of Microchip Technology Inc. in the USA and other countries. PIC is a registered trademark of Microchip Technology Inc. in the USA and other countries.

JUMPERS FOR PROTOTYPING AND R&D

Schmartboard has released a line of specialty jumpers which will give engineers an easier way to perform functions that sometimes require multiple hands. The uses are many, but in short, this product allows users to connect multiple points to a single common signal, save space, and reduce jumper use in tight areas such as ground.

It can also be used with electronic testing equipment such as logic analyzers as a signal interceptor to more easily access test points. Schmartboard includes a row of 40 headers for convenience. The 11" four-way bus jumpers will be available in five colors; pricing is \$9 for a set of five.

For more information, contact:

SchmartBoard

Web: www.schmartboard.com

BENCHTOP RF/MICROWAVE SIGNAL SOURCE

Berkeley Nucleonics has released a high performance, robust, and cost-effective high power microwave/RF signal generator. The model 845-H defines a new class of microwave source instrumentation capable of producing in excess of 23 dBm from 100 kHz to 20 GHz with very low phase noise, fast switching speeds, and extensive modulation capabilities in a portable lightweight benchtop package. With a compact sealed enclosure and a rechargeable internal battery option, this instrument offers a wide array of features to meet almost any application requirement in the lab, ATE, or field. The model 845-H — as well as all of BNC's microwave sources — are provided with a free intuitive GUI, as well as LabVIEW, C++, and MatLab drivers. Go to the website for custom pricing details.

For more information, contact:

Berkeley Nucleonics

Web: www.berkeleynucleonics.com

Recycling & Remarketing High Technology
WEIRDSTUFF®
 Warehouse
 Software, Computers, Electronics, Equipment, Doohickies
 WE BUY/SELL EXCESS & OBSOLETE INVENTORIES
 FREE COMPUTER AND ELECTRONIC RECYCLING
 GIANT 10,000 SQ. FT. AS-IS SECTION
 384 W. Caribbean Dr.
 Sunnyvale, CA 94089
 Mon-Sat: 9:30-6:00 Sun: 11:00-5:00
 (408)743-5650 Store x324
WWW.WEIRDSTUFF.COM

In the Nuts & Volts Webstore NOW!
NUTS AND VOLTS
 NV BOOK SPECIALS
 Programming the Raspberry Pi
 Getting Started with Python
 Simon Monk
 Raspberry Pi Three Book Combo
 Raspberry Pi User Guide
 Raspberry Pi Projects for the EVIL GENIUS
 Only \$48.95
 Plus FREE Priority Mail Shipping US Only
 To order call 800 783-4624 or visit:
<http://store.nutsvolts.com>
 Limited time offer

1. Apply solder.
 2. Listen.
boxed
 KIT AMPS
www.boxedkitamps.com
 DIY audio kits
 stereo [w] headphone [w] guitar

NATIONAL RF, INC.
 TYPE **RNF** HFDF
 NOISE LOCATION SYSTEM

 LOCATE NOISE IN THE HF SPECTRUM
 Visit www.NationalRF.com for this and other Radio Products!
 Office: 858-565-1319

NKC electronics
 THE OFFICIAL
 ARDUINO STARTER KIT!
NKCElectronics.com/starterkit
 Includes the Arduino Project Book (170 pages)
 Purchase Orders are accepted from Educational Institutions, US Government and Research Centers



SYNCHRONIZED LIGHT DISPLAYS



**A straightforward approach
for synchronizing Christmas
lights to music.**

Inspiration strikes — During the Christmas and Halloween seasons while watching random videos, you may have come across a crazy looking house with lights blinking and flashing synchronized to music. If you thought to yourself, "Wow! I would love to do that to my house," then this article is for you. With a little patience, this seizure-inducing fantasy can be yours, thanks to many off-the-shelf solutions available to illumination enthusiasts. I must warn you, however, if your neighbors or wife are not agreeable to flashing lights, hours of programming, and potentially hundreds of holiday revelers, then this may be your last chance to step away from an obsessive hobby. At the beginning, ask yourself what your goal is and work towards that. This will help you budget your time, money, and sanity accordingly.

In this article, I will be focusing on Light-O-Rama products, which is what I got started with. I initially chose the Light-O-Rama solution because this hobby can be very complex, and Light-O-Rama provides a straightforward approach to synchronizing lights that got me up and running in a short amount of time. I will also show how Light-O-Rama is compatible with different brands, allowing you to mix in other products so your light show can grow in scale and complexity.

Basic System Overview

From a system level, there are several components that make up a Light-O-Rama setup. **Figure 1** shows the typical light system. The general concept of the light display is that you have a PC at the heart of the system that sends out signals to light controllers to turn *channels* on and off so the lights blink on and off as programmed in the *sequence*. Each channel is an individually controlled element in the display.

The PC runs Light-O-Rama's S3 software. The computer runs scheduled shows; each show contains sequences, and each sequence includes a song (.wav or .mp3) and commands that are sent to the light controllers to turn the light channels on and off based on timings.

The PC connects via an RS-485 converter (USB-to-serial) to a network of Light-O-Rama controllers. The controllers connect to the RS-485 converter using a CAT5 cable, and then can be daisy-chained to each other using CAT5 cables.

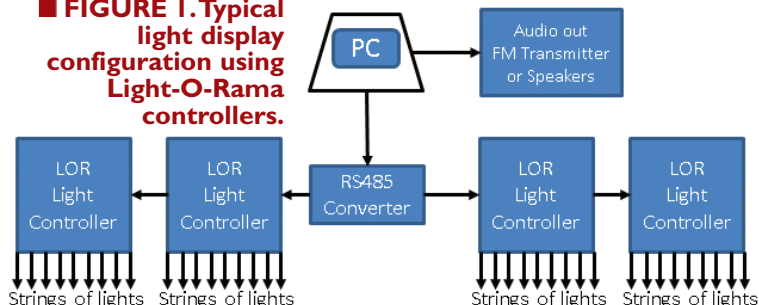
Light-O-Rama has a wide variety of light controllers. The most common is the CTB16PC (shown with its case open in **Figure 2**). These can be bought in kit form or fully assembled, and there are commercial grade versions, as well. The controller will plug into a power source and provide power to individual channels. The CTB16PC has 16 channels represented by 16 AC plugs coming out the bottom. Each of these plugs represents an individually controlled channel in the display.

A string of lights plugged into the channel will be turned on and off based on the programming in the sequence. So, one channel may be the lights around a window, another channel may control the lights on a wreath, etc.

This solution requires large amounts of extension cords (one per channel, plus for the controllers themselves). Making your own extension cords is easy using vampire plugs and SPT wire. This DIY method is cheaper than store-bought versions and is a great way to get custom length extension cords.

For audio, most people will connect an FM transmitter or speakers to the audio out port of the PC. Many build

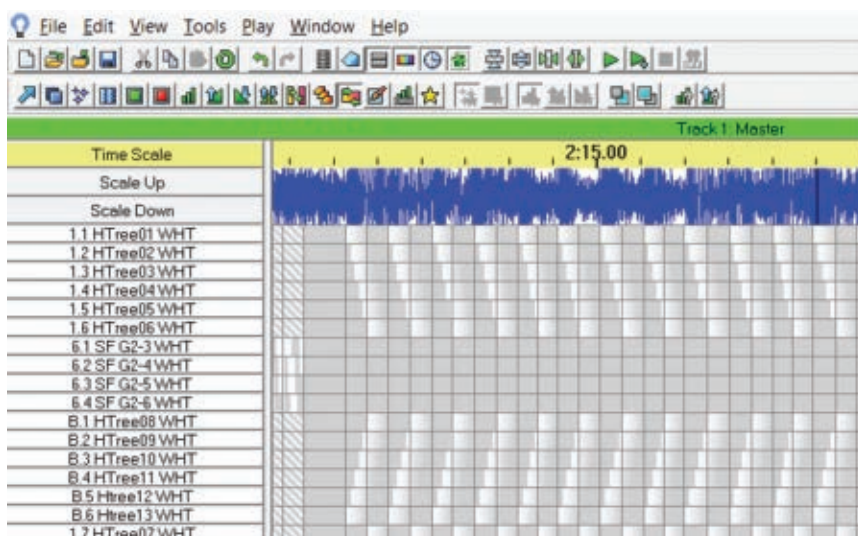
■ **FIGURE 1. Typical light display configuration using Light-O-Rama controllers.**



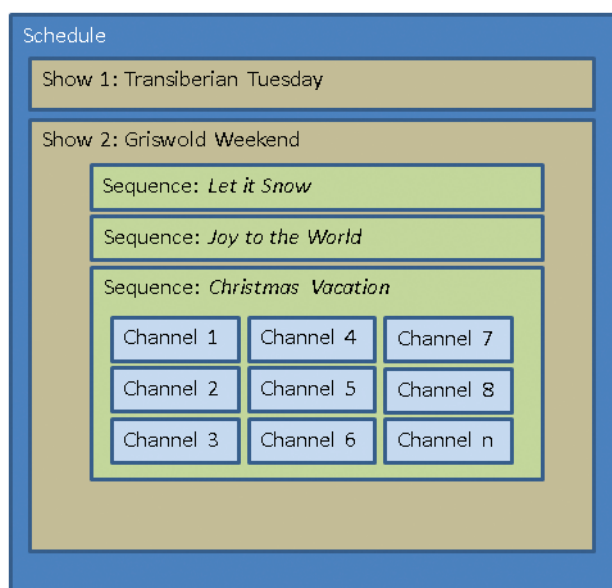
■ **FIGURE 2. Light-O-Rama CTB16PC light controller.**

their own transmitters from kits; others purchase transmitters online. The quality of transmitters varies greatly, so this is an area you will want to do your homework in.

For the past few years, the EDM kit has been a popular choice. People visiting the display will drive up, tune their radio to a specified frequency, and listen to the music while admiring your lights from the comfort of their car. This method also has the advantage of not blasting loud music which tends to disturb neighbors. The downside is you may get dozens of cars parked in front of your house, which will probably still annoy



■ **FIGURE 3. Screenshot of Light-O-Rama's S3 Sequence Editor.**



■ **FIGURE 4. Schedules contain shows that contain sequences that contain channels.**

your neighbors (even the nice ones).

Light-O-Rama S3 Software

Software is needed to program light sequences and to run the show. Light-O-Rama software operates in two modes:

1) A software development kit for designing your light display and programming sequences.

Light-O-Rama provides a Sequence Editor that helps you take a song and synchronize it to lights. First, the

software has built-in wizards that will automatically find the beat of the music and also automatically create timings that you can program to. You have the option of just tapping along with the music – then your tapping is converted into a timing grid. Since each sequence can have multiple timing grids, you can use multiple sets of timings to get the best result.

Using the Sequence Editor is much like using a spreadsheet (see **Figure 3**). The columns represent the timings and the rows represent the channels. You color in the cells to specify if you want the lights to be on, fade-on, fade-off, shimmer, or twinkle.

There is a definite art and a learning curve to sequencing songs, so plan to devote significant time to it. This is actually my favorite part, and I have online tutorials posted on YouTube that may be useful to you.

The S3 software also provides a visualizer that simulates how your display will look when programmed, without having to hang a single strand of lights. This way, you can develop all of your sequences in advance of putting the lights up.

2) Software that executes your light display schedule.

You configure a schedule that allows you to create an appointment for a show to start at a certain time and end at a certain time. A schedule can run the same show over and over, or run a variety of different shows that you create. Recall that each show contains sequences. The shows allow you to specify sequences that will start and end your show, and a list of sequences that loop until it is time to end the show. You can even have sequences that run in the background (**Figure 4**).

Currently, Light-O-Rama software only runs on PCs. System requirements grow as your display grows. Typically, running the show does not take a very beefy computer, but you may want some more processing power when programming the sequences and running the visualizer. Often, you will see people program their sequences on one computer, then run the show from a different one. In

Light-O-Rama offers sequences for sale on their website, but there is also an active Christmas Light community where people freely share their sequences. Everyone's display configuration is different, but with some copying/pasting and some gentle massaging, you can apply one of these pre-programmed sequences to your display vs. programming your own.

my display, I run the show using an Acer netbook with very little processing power.

The Creative Process

Synchronized lights is a great project for people that like to tinker. You do both hardware and software, and it's a project that can be done year round. This hobby does take considerable time, however.

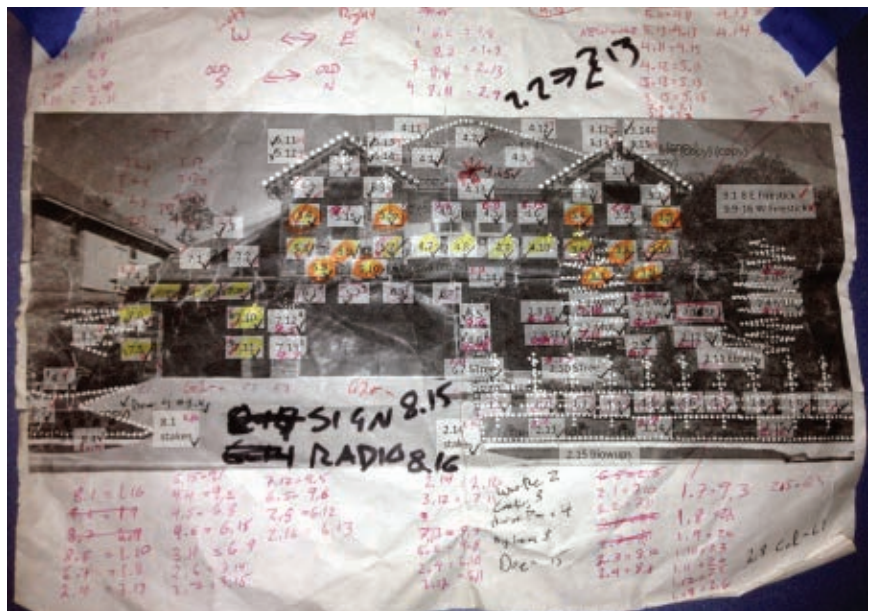
- 1) First, plan out your display. This is usually done after binge-buying Christmas lights in post-holiday sales. Figure out where everything will go and what needs to be returned to the store. Creating a light display around a theme produces good results. This step usually takes the form of a picture with lots of notes (**Figure 5**).
- 2) Create a channel configuration. Here, you configure what controllers you are going to use, what lights will be hooked up to each channel, and configure the visualizer. You will store the channel configuration Sequence Editor, export it to a file, and use it for each sequence. Each year, you will update your channel configuration, so you want to do a good job since you will import this configuration into each sequence you do. **Figure 6** shows my notes on mapping elements in my display to channels in my configuration.
- 3) Create a library of effects. Using your channel configuration, create a library of effects that you can copy/paste into sequences. This is a personal and creative process, but having a wave left-to-right and right-to-left is a great place to start. This is an optional step, but will make each song much easier to sequence.
- 4) For each song, you need to get a timing grid. You can use the built-in beat wizard, tapper wizard, and VU wizard to create timing grids. Even with the wizards, you will need to plan substantial time for each song because moving timings around to get it just right will take a while.
- 5) For each song, you need to program the lights. This is the super fun part. Here, you decide what each channel will do, when it turns on, when it fades, etc. Having the effects library to pull from (step 3) will greatly speed up this step.



■ **FIGURE 5. Planning where everything will go.**

light display, they think of the lights and sometimes the music, but there is another fun element that can be added: interaction. The Light-O-Rama controllers take a daughtercard, or you can buy an Input Pup that allows you to let people interact with your light show.

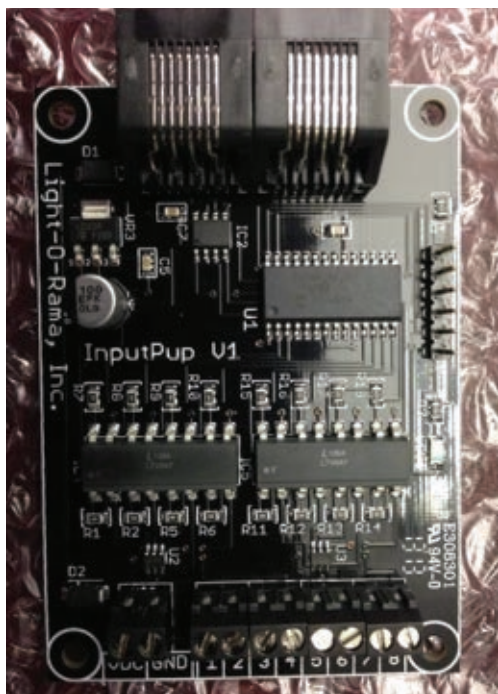
The Input Pup works off of simple circuits that — when closed — kick off a specified sequence. This can be used to rig a big red button that kicks off the show, let users pick the next song, or (with some creativity) be used to turn your light show into a video game that people can play.



■ **FIGURE 6. Notes mapping display elements to channel configurations.**

Making It Interactive

Normally, when people think about a



■ **FIGURE 7.** Light-O-Rama Input Pup used to trigger interactive events.



■ **FIGURE 8.** RGB "dumb" strip.

The best approach for making decisions is deciding on pixels vs. dumb, voltage (12 or five volts), type of lights (bulbs, strips, etc.), and topology. Based on those options, you then decide on which controller solution works best for your needs (**Figure 8**).

Light-O-Rama provides an off-the-shelf solution of RGB controllers and lights that sit on your Light-

O-Rama network. They also offer their SuperStar software to make programming them easier. For those wanting to do it themselves or to have more options, there are many other RGB light controllers on the market. Most RGB light controllers speak DMX and do not use the Light-O-Rama protocol, but Light-O-Rama software and hardware both speak DMX. This means that you can add non-Light-O-Rama controllers and still use the S3 software to program and control everything.

Figure 9 shows how you can have multiple light networks connected to your controlling computer. Here, I left the existing Light-O-Rama network in place and added a DMX interface to speak to the RGB controllers. This is a common configuration since many light displays will have a mix of RGB and non-RGB lights.

The DMX interface could be multiple DMX interfaces. One common interface is E.131 which is a DMX over Ethernet protocol, and the PC and RGB controllers speak

If the Input Pup is employed, then it is assigned a unique ID on your Light-O-Rama network just like the controllers (refer to **Figure 7**).

Making it RGB (Color Changing)

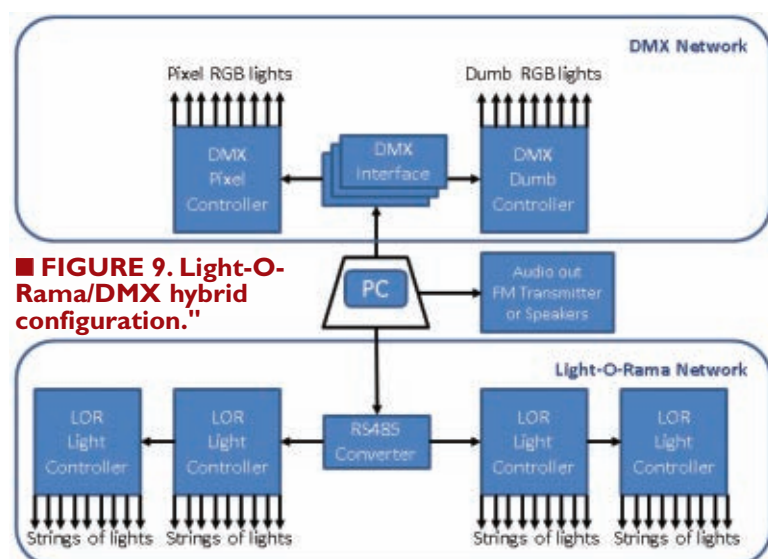
The big thing in light displays right now are RGB lights. These are color changing lights where you can set the lights to any color you can create by mixing red, green, and blue.

RGB lights are much harder to deploy than regular strings of lights. You can buy a prepackaged solution from Light-O-Rama, or you can purchase lights and controllers from China or small vendors. The latter solution requires some decisions and some homework to be successful.

There are two main types of RGB lights:

- 1) *Pixels* – Each light bulb can be independently controlled. Each bulb contains a microcontroller that watches for instructions and passes data on to the next bulb. These are often used on megatrees or matrixes, and can even display text and images.
- 2) *Dumb* – All the bulbs are controlled together. You can set them to any color, but they all change in unison.

What makes the addition of RGB lights challenging is the number of options available.



■ **FIGURE 9.** Light-O-Rama/DMX hybrid configuration."

to each other using IP over Ethernet. Another popular interface is using an Enttec Pro dongle that connects to the PC via USB, and then speaks raw DMX to RGB controllers behind it. Again, there are other choices and some research is needed to make sure you find the best solution for your configuration.

RGB lights have very specific power requirements. RGB lights are DC powered, so voltage drop is a constant concern. Often, power injection into the lights or signal repeaters are needed to make sure the RGB lights get the needed power and data to operate as expected.

Again, there are many solutions out there, so it requires some tinkering to get it right. If the Light-O-Rama solution is used, then these concerns are already addressed since you get the controller with the RGB lights as a packaged solution.

Conclusion

All of the above can sound pretty complex and challenging. However, while time-consuming, there are enough resources and products out there that make putting together that seizure-inducing light show a straightforward effort. The hobby keeps you engaged as you sequence more and more songs and incorporate new technology. If this is something that is interesting to you, I leave you with one tip of advice: start early. **NV**

RESOURCES

Light-O-Rama
<http://lightorama.com>

Enttec
www.enttec.com

EDM FM Transmitters
www.edmdesign.com

Audacity
<http://audacity.sourceforge.net>

XLights/Nutcracker
<http://nutcracker123.com/nutcracker/releases>

Online Community Resources:

Free sequences for download
www.highcountrylights.com/links/hcl-sequences.html

Light-O-Rama Forums
<http://forums.lightorama.com>

Planet Christmas
www.planetchristmas.com

Do-It-Yourself Christmas
<http://doityourselfchristmas.com>

My Resources:

My website — contains links to sequencing tutorials
<http://listentoourlights.com>

My YouTube Channel
<http://youtube.com/listentoourlights>

My Facebook Page
<http://facebook.com/listentoourlights>

Sequencing Tutorial
www.youtube.com/watch?v=Mf9O-KiysA0

BEST SCOPES, BEST PRICES



PASSPORT-SIZE PC SCOPES \$162+
Great scopes for field use with laptops. Up to 200MHz bandwidth with 1GSa/s, high speed data streaming to 1MSa/s, built-in 1GSa/s AWG/wfm gen. **PS2200A series**



30MHz SCOPE \$289
Remarkable 30MHz, 2-ch, 250MS/s sample rate scope. 8-in color TFT-LCD and AutoScale function. Includes FREE carry case and 3 year warranty! **SDS5032E**



50MHz SCOPE \$399
50MHz, 4-ch scope at 2-ch price! Up to 1GSa/s rate and huge 12Mpts memory! Innovative "UltraVision" technology for real time wfm recording. FREE carry case! **DS1054Z**



60MHz SCOPE \$349
Best selling 60MHz, 2-ch scope with 500MSa/s rate plus huge 10MSa memory! 8-in color TFT-LCD. Includes FREE carry case and 3 year warranty! **SDS6062V**



70-300MHz SCOPES \$839+
Fast, versatile 2-ch 2GSa/s scopes with 8-in WVGA LCD, integrated generator, 14Mpt memory, very low noise floor. FREE carry case available! **DS2000A series**

FREE TECH SUPPORT - GREAT CUSTOMER SERVICE **Saelig**



SF02 Laser Rangefinder

Fast and accurate distance measurements from 0-40 m (0 to 130 ft). Not affected by wind, noise, ambient light, terrain, air temperature, or changes in barometric pressure. *No calibration required* - ready to use right out of the box. **#28043; \$349**

Order www.parallax.com
or call 888-512-1024
(M-F, 8am-5pm, PDT)

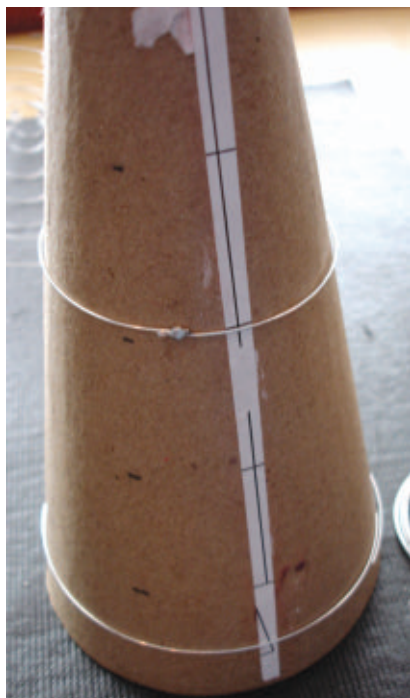
PARALLAX
www.parallax.com

BUILD THE 3D CHRISTMAS LED TREE

I've been married for 53 years now, so finding my wife a Christmas present becomes more of a challenge. It's not like I can't afford to go to the store and buy something. It's just that I was raised believing that a gift should come from the heart and be something handmade. I'll never forget when I was eight, my dad made me a plywood jeep with a 30 caliber machine gun mounted in the back that was made out of a cardboard shipping tube.



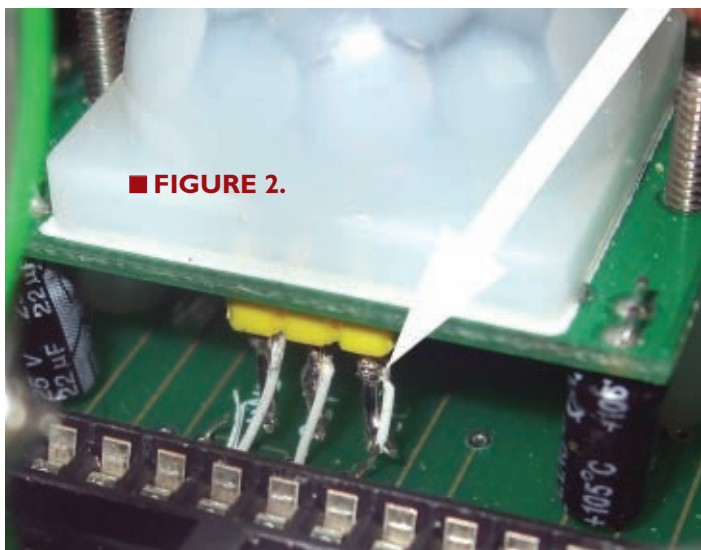
■ FIGURE 1.



Due to allergies in the family, we can't have a live Christmas tree, and I'm getting too old to put up a fake one. My grandkids are all older too, so now I'm waiting for great-grandkids. Thumbing back through some old issues of *Nuts & Volts*, I came upon an issue that had a Christmas tree made out of LEDs. Searching the Internet, I found several methods for making a cone shape tree utilizing different techniques. While there are a plethora of kits available, I decided to come up with my own (**Figure 1**). I added a pyroelectric infrared (PIR) sensor for a proximity detector, along with a microphone circuit which detects music and then flashes accordingly to the beat.

The tree has a four inch square base and a height of 10.5". There are eight columns which have 56 LEDs. The top LED is a tri-colored unit that changes colors. The LEDs are multiplexed to save power, and appear to be on all the time even though they're not.

Let me digress. In 1948, Southern California Edison switched from 50 Hz to 60 Hz. I remember this well as all the electric clocks had to be discarded, along with many motors. Using 50 Hz and incandescent bulbs, no one could see the flicker since the filament could not cool off that fast. Keep in mind that when using a 50 Hz alternating current, there is a period of time (100 times per second/zero crossing) when the power is off. However, in 1938 when fluorescent bulbs became popular, a noticeable flicker could be observed that drove people nuts.



■ FIGURE 2.

PIR MODULE MOUNTING

Turn the PIR module over so that the pots are on the top of the board. To the upper right are three pads: one with an "L;" one unmarked; and the other with an "H." Take a razor blade and cut the trace to the center pad and the H pad. Using a small wire, short the L and the center pad. This will cause the PIR to fire every second when there is a body present triggering the tree.

On the PIR board, there are three pins: "GND," "OUT," and "VCC." Solder three wire-wrap wires to these pins and strip the insulation off of each one. Thread the three wires into the main board. Place the two 2-56 3/4" screws through the bottom of the board and add the two spacers. Thread the screws into the PIR board. They should self-thread. Solder the three wires (**Figure 2**).

MAKING CHRISTMAS HULA HOOPS

The hoops of the tree are made out of 18 gauge bus wire, which is quite malleable. Go to the article link and download the hoop template. Cut the bus wires to the ring sizes under the template. Bend the bus wire to fit (**Figure 3**). Butt the ends of the wire together and solder. I'm a firm believer in using "tacky flux" by Quick Chip© and always have a syringe around for both soldering and de-soldering. Massage the wires so they are flat and round. There are seven hoops; do the same for the rest.

With each hoop, place the butted ends between the spokes. Place a drop of solder on each spoke crossing. At first I tried using a Sharpie©, but it makes a big black mess with the tacky flux on the LEDs.

Cut eight bus wires to a length of 10". Using two pairs of pliers, pull the wires taut and run them over a sharp edge (of a table, for example) to straighten them.

MOUNTING THE LEDs

I found it best to mount the 10" wire on a ruler with two pieces of tape at the ends; refer to **Figure 4**. If you are using colored LEDs, download the LED matrix file. The microprocessor is programmed to the color pattern on this matrix. (I used one of wife's muffin pans to put the LEDs in so they didn't get mixed up.)

Bend the long anode lead of the green LED out at a right angle. Slide it under the column wire with the LED pointing to the right. Solder the **anode** to the wire so there is about 1/16" between the LED and the wire at position 1 (which is 1-1/4" from the end of the wire).

Using the matrix, go



■ FIGURE 3.

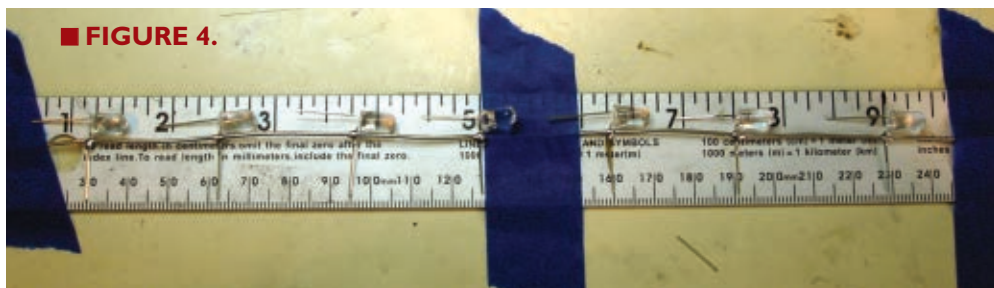
220 ohm resistors and place them into R1-R8. Bend the leads of nine 10K resistors and place them into their respective holes in R9-R17. Place a 100K into R18, 27K into R20, and a one megohm into R21. Solder all the resistors.

Solder R19

which is the pot. Solder C1; it has no polarity. Solder C2; it does have polarity. Put the long LED into the + pad. Hold the microphone so it's facing its pins, and rotate it so the pins are at the top. Pin 1 is to the left and is positive. Solder the microphone. Solder the two switches and transistors, noting their outlines on the board.

Run the two battery wires from the battery holder through the strain hole below C1; solder the black wire to "B" and the red one to "R."

Mount the battery pack to the bottom of the board using double-sided tape or just put it to the side. (If you put the battery holder to the side, add four rubber feet.)



■ FIGURE 4.

■ **FIGURE 5.**



DOWN the column for the next colored LED, which will be blue, then yellow, then red. Space each LED at 1.25" (e.g., 1.25", 2.5", 3.75", 4", 5.25", 6.5", 7.75", 9"). Cut off the excess anodes, and put a piece of tape on the finished column with the #1 on it (**Figure 5**). Do the same for the next wire, noting that the first LED is blue; label it #2. Do the same for the remaining six columns.

Put column 1 into a vice with the LED pointing up. Place the first ring with its first mark on the cathode and bend the cathode up to hold the ring. Solder the cathode to the ring. Continue for each ring until all of them are soldered.

Locate column 5 and mount it in the vice with the LEDs pointing upward. Locate spoke 5 on the rings, which will be 180 degrees from column 1. Thread the column through the rings and solder the LEDs to the ring. This procedure will give structure and will make it easier to keep the form and symmetry of the cone (**Figure 6**).

Going clockwise from column 1 (looking from the top), solder column 3 and column 7, then column 2 and 6, and finally columns 4 and 8. Re-solder any LEDs that are out of alignment.

THREE-COLORED LED

Cut the top wires off of columns 4, 5, 6, 7, and 8 to their last LED. You should now have three wires from the top of the cone. Cut an 11" green wire-wrap wire, wire wrap it to the long LED lead (cathode), and solder. You may have to cut it shorter so that the wire is next to the LED. Going left to right, lead 1 is green, 2 is blue, 3 is the cathode, and lead 4 is red.

Solder lead 1 from the LED to column 1 so that the three-colored LED is in the center of the cone as the last layer of LEDs. Solder LED 2 to column 2 and lead 4 to column 3. Cut the excess wires off.

Take the green wire from the cathode of the three-colored LED and wrap one turn between each LED on column 8. Leave about 1" below the column wire for soldering to the board.

Locate column 1 and solder a green wire to ring 1, then wrap one turn around column 1 with 1" below the column.

Using column 2/ring 2, solder a green wire. Continue to ring 7, wrapping one turn around the columns between the LEDs. Refer to **Figure 7**.



■ **FIGURE 6.**

ITEM	DESCRIPTION	MODEL	QTY	SOURCE
BH	Battery holder 3 AA	534-2465	1	Mouser
C1	.1 μ F	594-K104K15X7RF5UH5	1	Mouser
C2	10 μ F	581-TAP106K016CRW	1	Mouser
IC1	Proximity sensor	RB-LTE-116	1	Robot Shop
Mic	Microphone	665-AOM4546PNFR	1	Mouser
LED 1	Red	859-LTL353QRKNN	14	Mouser
LED 2	Amber	LTL2R3F859-LTL2R3YW3KSV3KS	14	Mouser
LED 3	Green	859-LTL3H3TGUADS1	14	Mouser
LED 4	Blue	859-LTL2P3TBU2KS	14	Mouser
LED 5	Multicolored common cathode	RL5-RGB-DCC	1	SuperBrightLEDs
R1-R8	220 ohm 1/4 watt	291-220-RC	8	Mouser
R9-R17	10K 1/4 watt	291-10K-RC	9	Mouser
R18	100K 1/4 watt	291-100K-RC	1	Mouser
R19	10 pot	652-3309P-1-103	1	Mouser
R20	1 meg 1/4 watt	291-1M-RC	1	Mouser
R21	27K 1/4 watt	291-27K-RC	1	Mouser
Screws	2-56 x 3/4"	534-9357	2	Mouser
Socket	Socket 28-pin	526248	1	Jameco
Standoff	3/8" 2-56	761-1111-2-N	2	Jameco
SW1	On-Off	612-EG1218	1	Mouser
SW2	Momentary switch	612-TL1105-250	1	Mouser
U1	PIC16F916	PIC16F916-I/SP	1	Mouser
U2	Op-amp	579-MCP601-E/SN	1	Mouser
U3-U10	NPN transistor	512-PN222ATF	8	Mouser
Wire	18 bus	2098507, 12'		Jameco
Wire	Green wire wrap	22606, 10'		Jameco

PARTS LIST

MOUNTING THE TREE

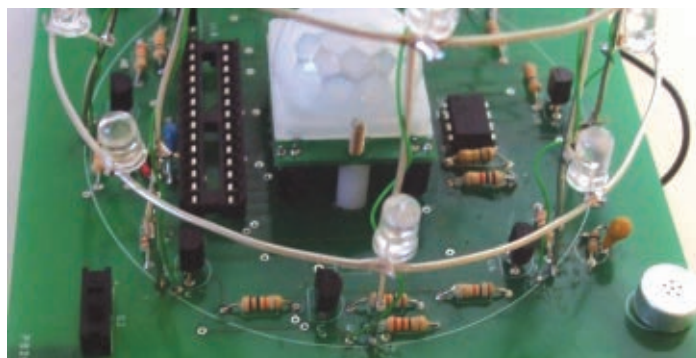
Starting with column 1 (the nine o' clock position) and going clockwise, push the columns through their respective holes. Make sure that the tree is vertical and not looking like the "Leaning Tower of Pisa." Solder the columns, then solder the wire-wrap wires to the hole labeled "W." Push the columns into their holes which are numbered. Again, make sure the columns are straight. Solder all the columns. Place the wire-wrap wires into their holes next to the column holes marked W and solder. Trim off any excess wires so it looks like **Figure 8**.

USING THE TREE

Put in three AA batteries, noting their polarity. Turn on the slide switch. The tree should light up, giving many different patterns. There are actually four different modes that can be changed. If you just turn the unit on, it will continuously run until turned off. Hold down the momentary switch while turning on the power. Then, when the switch is released, the top LED will turn blue, indicating that you are in microphone mode. By pushing the momentary switch again, the top LED will turn green, indicating you are in the PIR mode. Push the switch again, and the top LED will turn red; all the LEDs will remain lit. There is a built-in 2-3 second delay that allows you to change settings. Keep in mind there may be slight delays due to the capacitor charging before the PIR and microphone start working.

PIR (GREEN LED)

When the PIR module detects a change such as body



■ **FIGURE 7.**



heat, the tree will light for a period of time, then shut off. It will continue to cycle if there is any movement.

There are two adjustments on the PIR module. When mounted (viewing from the back), turn the left pot all the way counter-clockwise. This is a delay which holds the pulse high for a period of time. Turn the right pot all the way clockwise. This is the sensitivity. There is a 2-4 second delay before the PIR will trigger.

MUSIC BEAT (BLUE LED)

Place the tree next to a speaker or on top of one. The light will change color when sounds are detected. You can also clap to activate the microphone. The sensitivity can be changed by varying R19; going counter-clockwise will increase the sensitivity.

ALL ON (RED LED)

The LEDs are being multiplexed and appear to be on all the time (even though they're not).

HOW IT WORKS

The columns (anodes) are driven by port B of the microcontroller; the ports can sink or source up to 35 milliamps. Port C connects each of its ports to eight NPN transistors via 10K resistors, which pull the LED cathodes to ground. A positive voltage on the NPN transistor will cause it to conduct.

The patterns are generated by the microprocessor mainly by rotating the bits of ports B and C right or left, and turning the LEDs on and off for short periods of time. The assembly files are available at the article link for those interested. I have added a socket so you can create your own programs and experiment around if you'd like.

The PIR and microphone are directed to the interrupt vectors of the microprocessor. The microphone is amplified 100 times by an operational amplifier (U2), and when a voltage pulse is detected the microprocessor jumps to the random generator. In the interrupt mode, the microprocessor is put to sleep to save power.

Have a Merry Christmas and be the LED of someone's life! **NV**

■ **FIGURE 8.**

GET BUILDING WITH

Actobotics

DREAM. DESIGN. BUILD. REPEAT!

BRAND NEW

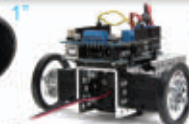
BROUGHT TO YOU BY

SERVOCITY

SHOP ONLINE
OR CALL 620.221.0123



1/2"
Carbon
Fiber
Tubing



The ActoBitty
Robot Kit
\$27.99



Channel Snap
Mounts



Stepper Motor
Mounts
\$5.99 - \$7.99



32 Pitch Beam
Gear Rack



Battery Mounts
\$1.79 - \$3.99



Brushless
HS-93XXTH
\$179.99



HS-8385TH
\$147.99



HS-70MG
\$31.99



Dongbu
Servo Gears
\$14.99



Metric
Bearings



Wheel Brackets
\$7.99



Bevel Gears
\$5.99



NEW!
Off-road
Kits!



Start your
project off right
with our new robot
kits!

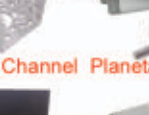


Coca-Cola



Coca-Cola

TRIED & TRUE

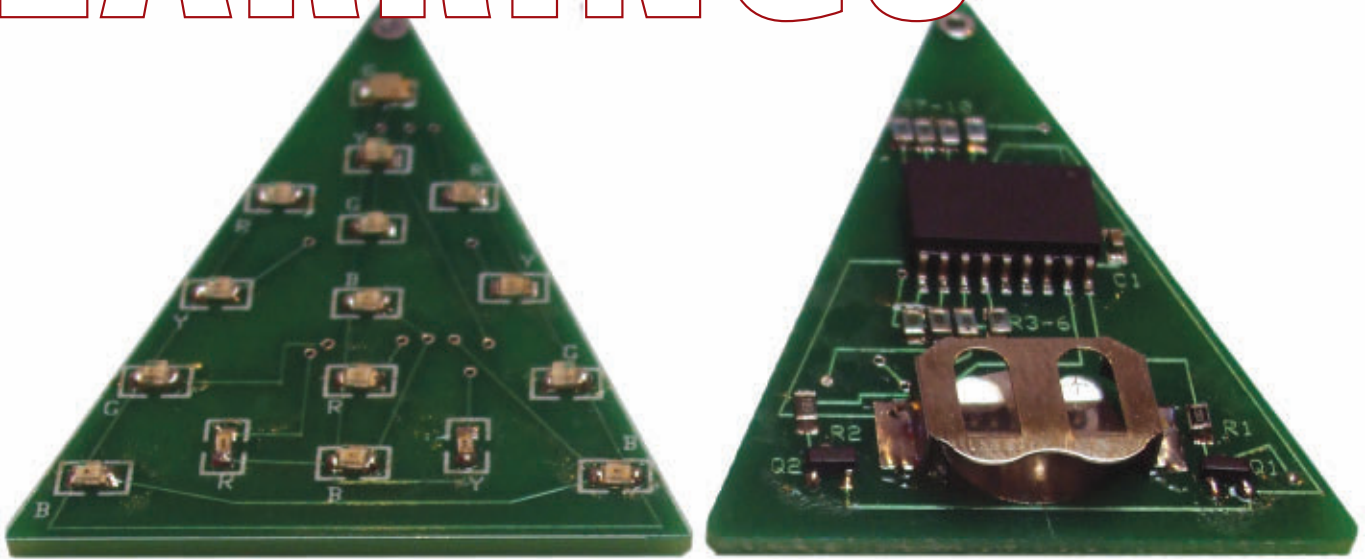


FIND US ON



SERVOCITY.com

MAKE THESE UNIQUE XMAS EARRINGS



This is not only a great teaching project for soldering surface-mount components, it is also perfect for the holidays and very inexpensive. Now, you can help cultivate interest in a budding student getting started in electronics AND take care of some gift giving in one fell swoop.

There are two projects in this article. The first utilizes surface-mount parts with a microprocessor. It has 16 different colored LEDs that flash randomly, blink, have patterns that run up and down and down and up, and flash different colors. It equates to about one minute of programming. The other project uses through-hole components. The new rainbow LEDs used here change colors and blink. This is about a three minute project. (My wife actually prefers this style.) Both versions use a small hearing aid battery. There's a video available in the NV Webstore if you'd like to see these two different types in action.

This project helps to teach hand surface-mounting and has assembly files that can be changed to make your own patterns. Normally with my projects, I include programming pads so you can change the programming on the board. This one does **not**, however. The low impedance on the programming pathways prevents this,

plus there's not much space.

The board files were generated using ExpressPCB and their free software (www.expresspcb.com). These files along with hex and assembly files for the microprocessor are available at the article link. If you don't want to worry about programming, there is a complete kit available from the NV Webstore.

Okay ... let's get started by first discussing some helpful tools and tricks to make your construction faster, safer, and easier.

You won't really need much in tools. Curved forceps (check out Mouser 578-EROP7SA for \$3.90), a small tipped soldering tip (I use a Weller PTS8 1/64" long), "Tacky Flux" (Chip Quik SMD 290), small solder wire (I use Kester 0.60 mm), and solder wick .030 (Digi-Key 60-1-5-ND). If you're like me, a lit magnifying lens, loupe, or head magnifier will also be of use. Check out **Figure 1**.

You won't need a reflow oven for this project.



SIMPLE TOOLS THAT HELP

USE AN APRON

Although you may look weird, it is well worth using an apron. Find a white or neutral color with no pockets. **Do not use** an apron with a busy design in case a component decides to fall on you (or the floor). Pick up some Velcro® and sew or glue it on the top side of the apron in the two corners and also in the center of the bottom edge (I usually attach three squares of Velcro.) Refer to **Figure 2A**.

Stretch the apron out, and put three 4" strips under the edge of the table where the Velcro strips are. This will allow flexibility when using the apron (**Figure 2B**).

Now, if something decides to fly off your work area, it will be easier to retrieve it.

MAKE UP NEEDLE PROBES

Nothing is more frustrating than to find a surface-mount resistor or capacitor that has gotten mixed up and has no markings. Make up a set of probes using two 1.5" needles. Solder the needles to a small length of flexible wire and tin the other end. (This is becoming more of a sewing article than an electronics one!)

When you find a strip of components and don't know what they are, you can pierce the package and measure. This also makes it easier to identify the anode and cathode of LEDs or diodes if the markings have rubbed off.

HOW TO SURFACE-MOUNT BY HAND

One of the tricks to surface-mounting is to have a very steady hand and a board that will not move. I use double-sided tape. This will allow your forearms to rest on the work bench. It is also great for sticking battery supplies to another chassis.

Place the double-sided tape to the bench where your arms will feel comfortable. Have it extend farther than the board. Peel off the upper side of the tape and stick the earring board to it. This makes it easy if a part is upside down or the marking is on the wrong side. Just touch the part to the tape and (it will stick) roll it.

Let's try this with a green LED. Find the pads that have a "G" and add a dab of tacky flux to both pads. Find a green colored LED, locate its cathode (normally has a marking on top), and place the cathode next to the G marking. On the Christmas tree board, the cathode is marked with the color of the LED. Solder the components so they are horizontal and not vertical. (If they're vertical and you are right-handed, your tweezers "hand" will be in your stomach.) Watch the LED cathodes on the right side as they are opposite of those on the left side and center. (Refer to **Figure 3**.) Remember to rotate the board, not the components.

■ **FIGURE 1.**



The first thing I do is see if the board pad has enough solder to tack the part. Most boards are tinned and many components are also. Put a small dab of tacky flux onto the pad and place the part onto it, holding it with the



■ **FIGURE 2B.**

■ **FIGURE 2A.**



curved forceps. Touch the soldering tip to the pad and component. If it sticks, you are home free. If not, add a small amount of solder to the pad and try again. Once tacked, use sparse amounts of solder that touch both the pad and the component's connector. If you put too much solder, you'll create a solder bridge! Just take the clean end of a piece of solder wick and touch it to the part. Ta-Da ... it's cured!

Some quick rules of thumb: 605s are very difficult to hand solder; 805s are fairly easy to hand solder; and 1206s are a piece of cake.

THE "DAUNTING" CHIP

Actually, chips are really easy. Place tacky flux on each pad and place the chip onto the pads. Make sure pin 1 is in its correct position! Remember: Rotate the board, not the components.

Tack pin 1 and the pin in the upper right-hand corner. Solder the rest of the pins. Some people flood all the pins by dragging the solder across them, then remove the solder bridges using solder wick. This is up to you, of course, but if you're a novice, I would practice soldering

each pin individually.

LET'S BUILD EARRINGS

Now that I've made you an expert in hand soldering surface-mounts, let's complete the board.

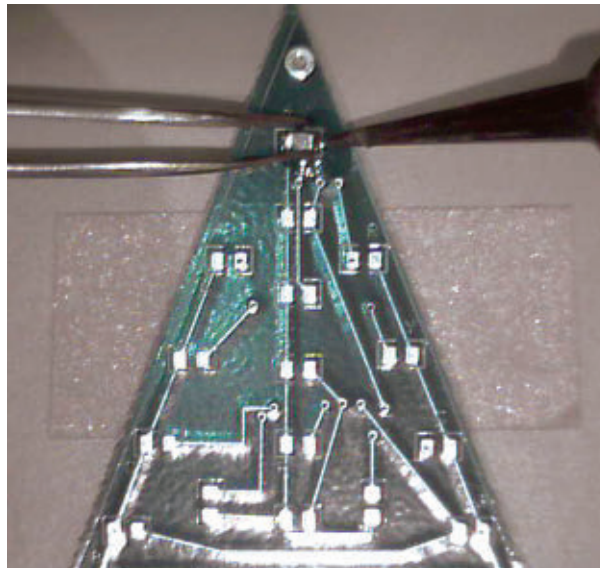
To keep track of all the colored LEDs, I would mount all the greens at the same time, following with the other colors. Dump all the greens into a canister (or receptacle of your choice).

Solder all the green LEDs, then the red, yellow, and blue ones to the top of the board.

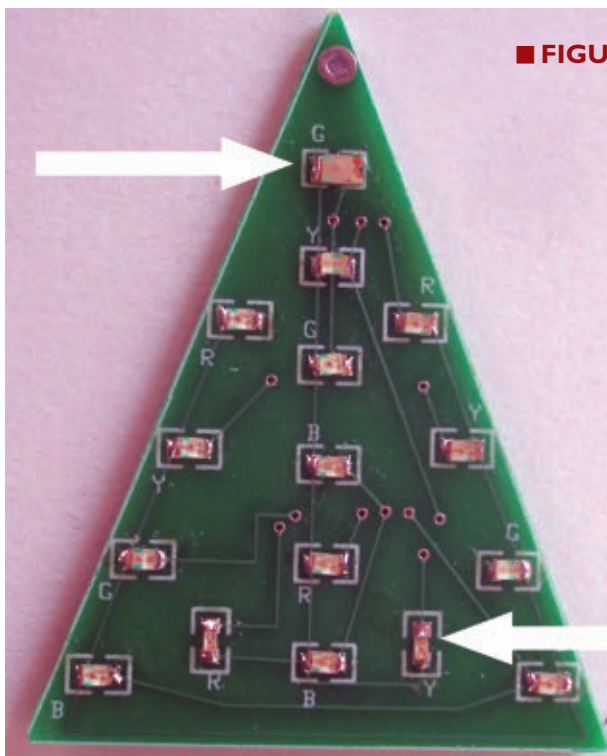
Remember the cathode goes to the color marking. Only LEDs will be on top of the board. Note: For comparison of sizes, I soldered a 1206 LED for the top and two 605s on the vertical LEDs just to show you can actually hand solder 605s. The rest are 805s (Figure 4).

The reason we did the top first is that it makes a flat surface when turned over.

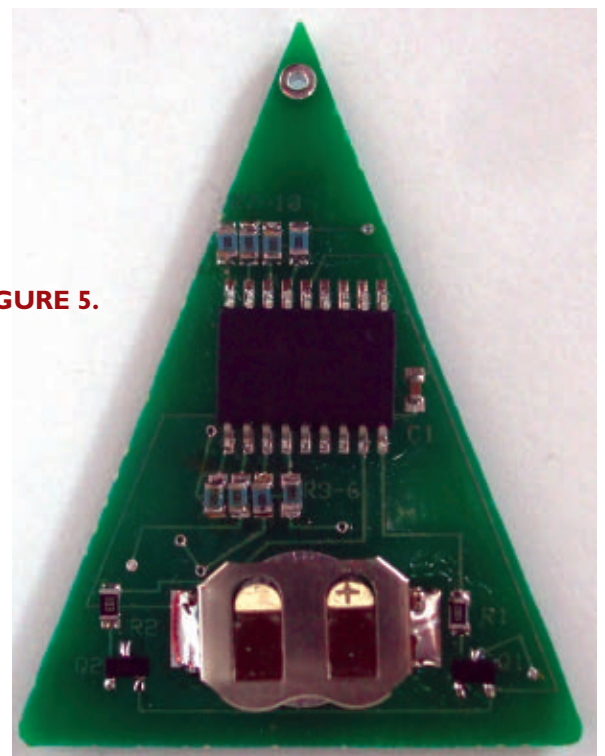
So, turn the board over and place the top of it to the double-sided tape. Solder the two transistors, U2 and U3. Solder the two 10K (R1-R2) resistors just above the



■ FIGURE 3.



■ FIGURE 4.



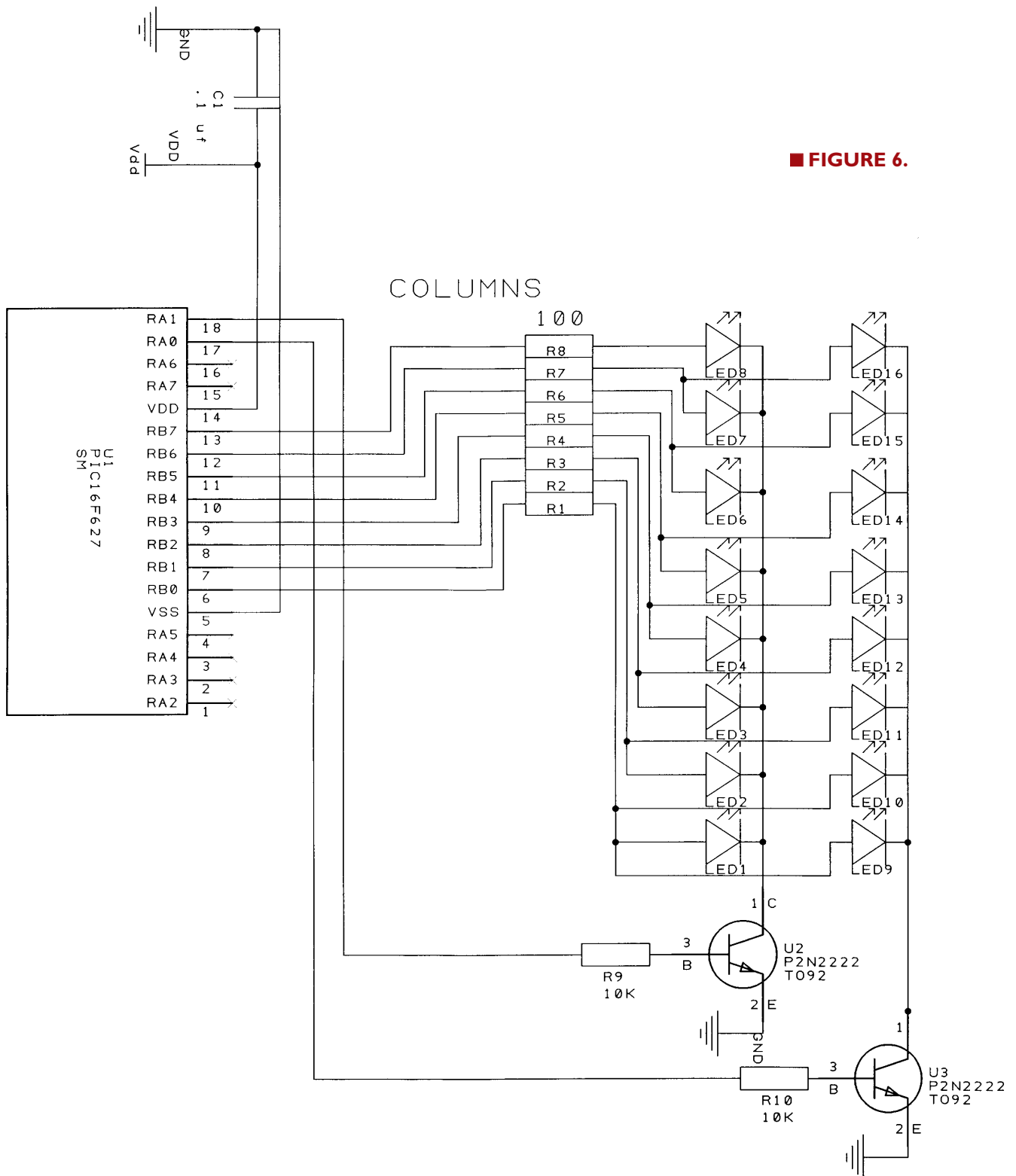
■ FIGURE 5.

transistors. They do not have any polarity.

Solder the remaining 100 ohm transistors (R3- R10) to their pads. Solder C1. It has no polarity, either. Solder IC1, noting where pin 1 is. Solder the battery clip (**Figure 5**).

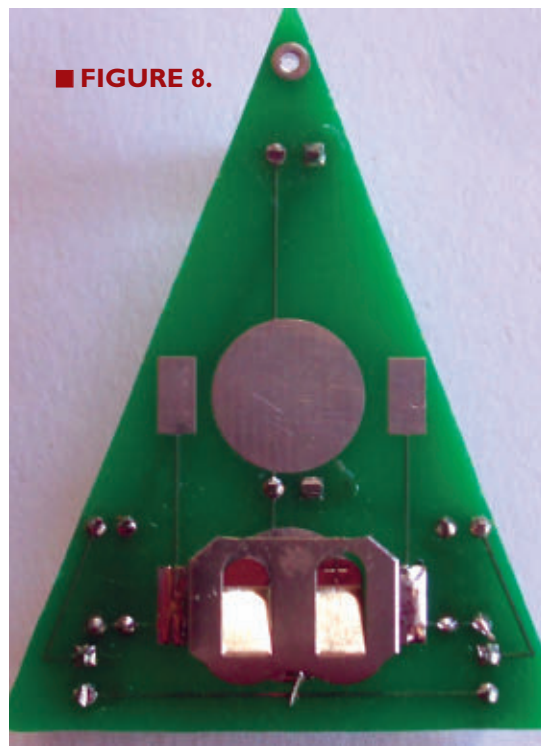
Put in the CR1225 battery, and the earring should start to flash. The battery should power the earrings at least eight hours. I ran a one minute statistical analysis of both units, and the average draw is about 2-3 milliamps.

■ FIGURE 6.





■ FIGURE 7.



■ FIGURE 8.

THE CIRCUIT

The Microchip PIC16F627 used here is a low cost microprocessor. It has its own built-in clock, and is capable of sinking and sourcing 25 mA, and runs from 2 to 5.5 volts. A three volt lithium battery was used in this circuit for power. The clock is set to run at 4.0 MHz. Port B provides anode current via 100 ohm resistors. The cathodes are pulled to ground using two NPN transistors with a base resistor of 10K.

To turn on the transistors, a positive voltage is applied to the base. Two columns of eight LEDs are used, with each column having a common cathode. There is a possibility of 512 different light combinations ($2^9 = 512$), but I am only going to use a few (my fingers get tired of typing code, as well as my brain). The lights are multiplexed to save power and give the effect of being on all the time (**Figure 6**).

To give a simple example of the code, you can turn on all the anodes on one side of the earring with a binary setting of 11111111, or a hexadecimal of 0xFF, or a decimal of 255 on port B. All the anodes will be on.

Placing a binary setting of 00000001 or 0x01 in hexadecimal, or just 1 in decimal on port A will turn on column 1. If you just want to turn on one LED — for example, the first one — writing a 1 to port B and a 1 to port A will turn it on. If you want to turn on LED 9 (the other column, first LED), writing a 1 to port B and a 2 to port A would turn it on.

Since the microprocessor is running so fast, time delays have to be added to slow down the on and off times. The schematic is shown in **Figure 6**.

SIMPLE SOLDER EARRINGS

Okay, say you need some bling in a pinch. Push the four resistors through the top of the board and solder. Push four 3 mm LEDs through the top of the board, putting the long leads into the square pads (**Figure 7**). Turn the board over and solder the battery holder. You just surface-mounted (**Figure 8**)!

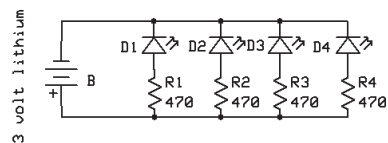
Put a battery in and the earring should light.

FINAL COMMENTS

This circuit takes advantage of color-changing RGB LEDs. Unlike conventional LEDs that require external intelligence to make them blink or fade, these RGB LEDs have a microscopic microcontroller encapsulated within them. Subsequently, you only need to add as little as two volts of power to have a colorful fading/flashing light show (**Figure 9**).

Have a Merry Christmas. **NV**

■ FIGURE 9.



ITEM	DESCRIPTION	PART #	QTY	SOURCE
BH	12 mm battery holder	712-BAT-HLD-012-SMT	1	Mouser
U1	PIC16F627 SM	579-PIC16F627A-I/SO	1	Mouser
LED 1	Green	859-LTST-C170GKT	4	Mouser
LED 2	Yellow	859-LTST-C171KSKT	4	Mouser
LED 3	Red	859-LTST-C171CKT	4	Mouser
LED 4	Blue	859-LTST-C171TBKT	4	Mouser
U2-U3	NPN surface-mount	863-MMBT3904LT1G	2	Mouser
R1 - R2	10K	RK73H2ATTD1002F	2	Mouser
R3 - R10	100 ohm 805	RK73H2ATTD1000F	8	Mouser
C1	.1 μ F	C0805C104K5RACTU	1	Mouser
Ear wires			1	Hobby store
Batteries	Battery = CR1225	614-CR1225.IB	1	Mouser
Board				
SIMPLE SOLDER EARRINGS				
BH	12 mm battery holder	712-BAT-HLD-012-SMT	2	Mouser
Battery	Battery = CR1225	614-CR1225.IB	1	Mouser
Ear wires			1	Hobby store
LED 1-4	Fast Changing	Rainbow LEDs 3 mm	4	
R1-R4	470 1/8 watt resistor	299-470-RC	4	Mouser



For all your product and project graphics solutions.

4D Systems designs, develops and manufactures intelligent graphics solutions using the latest OLED and LCD technology available, with custom graphics processors that enable both stand alone and host dependent solutions suitable for a very wide range of applications and projects.





4D SYSTEMS
TURNING TECHNOLOGY INTO ART

www.4dsystems.com.au
■
[f.com/4DSystemsAU](https://facebook.com/4DSystemsAU)
■
[t.com/4DSystems](https://twitter.com/4DSystems)

WISHING YOU A VERY RANDOM CHRISTMAS



This flexible eight-channel light controller uses random number generators to make sure your show is never the same.

For the past few years, I have had as part of my backyard Christmas light display a 'tree' made up of eight LED light strings strung in a pyramidal fashion to a central metal pole. (Refer to “The Tree” **sidebar**.) At night, it was a nice display that worked well, but as a static display it grew to be boring. I needed to add some action using a light controller.

There are a number of commercially-available Christmas light controllers that can be programmed to act in conjunction with music, control many channels of lights with various effects, and handle high current loads — all with price tags that match the capabilities. Other options are simple, but repeat the same pattern and quite often are integrated within the light string itself. For me, I didn't want to have to synchronize with music; I did not want the same patterns night after night or something that required operator interaction. I just wanted a simple controller that I could plug in a few sets of lights to and apply power.

The project in this article is low cost and simple to build, while providing several different light patterns in random order and speeds. Each time the circuit is powered up, the pattern order is different, only repeating after 65,535 power cycles. Several of the patterns are random themselves, and the speed of light change is randomly selected. Should you want to, it is easy enough to change the speed and modify the patterns.

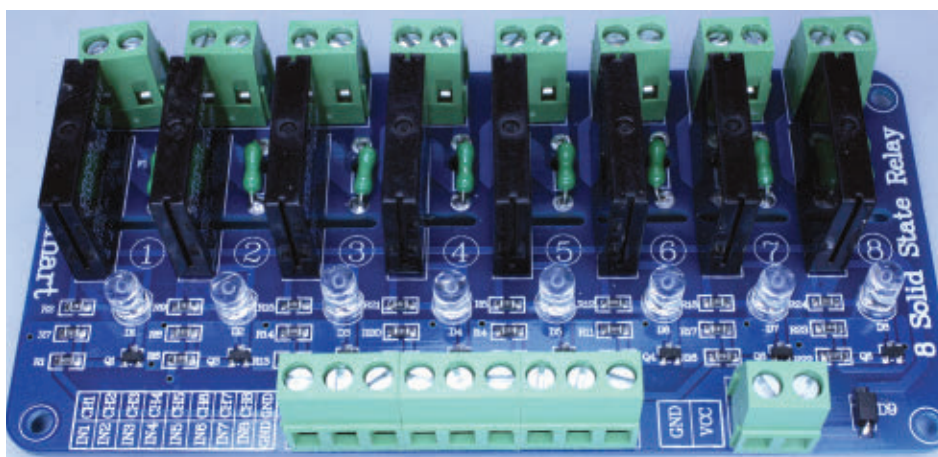
Circuit Description

The circuit is built around the PIC16F690 and an eight-channel solid-state relay (SSR) board that is readily available. The circuit (as shown in **Figure 1**) is simple enough to build on a perf board. Configuring the board so it interfaces to the SSR board was also fairly easy. If you use a different relay board, then you may have to modify it.

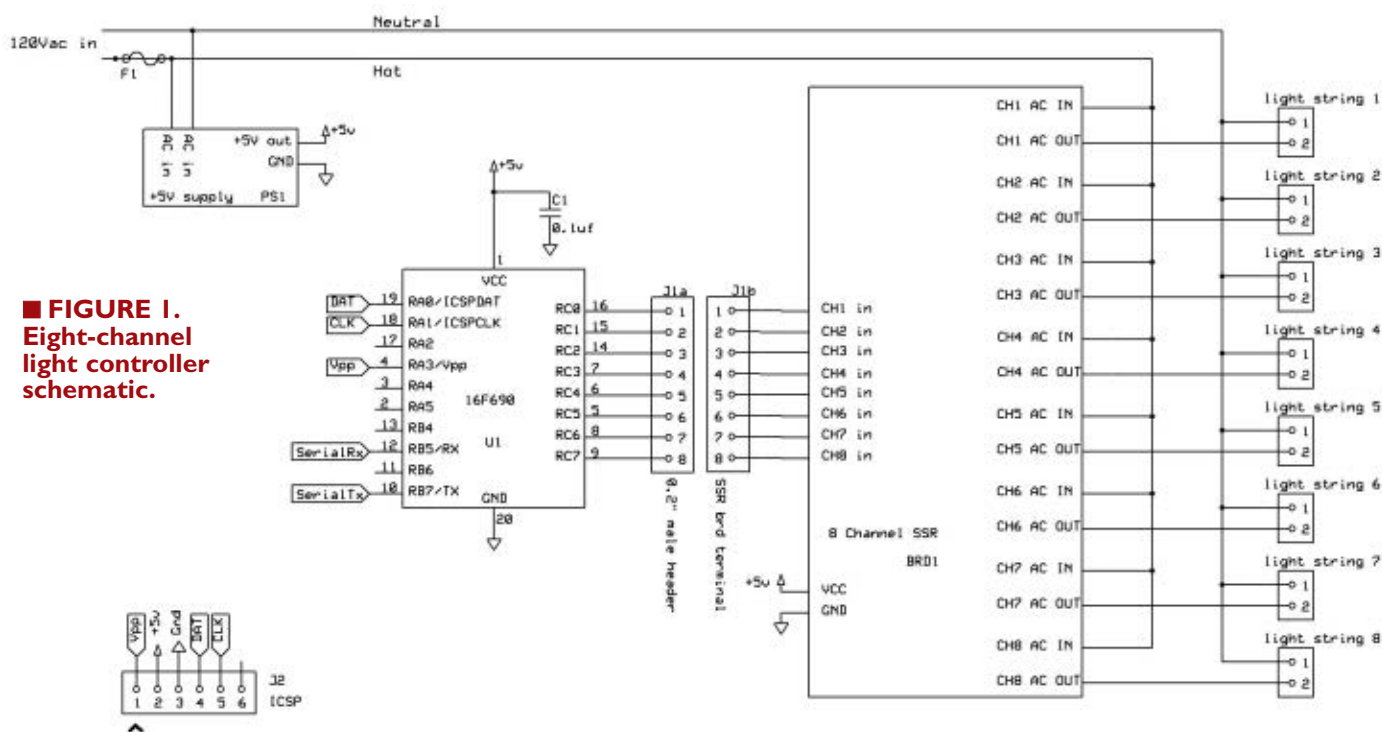
Though I selected the PIC16F690 for this project, there are many other eight-bit microcontrollers that could be used. I chose the 16F690 because its port C is eight bits wide, making it very easy to address as we will see later in the description of the software; there are enough other I/O pins to leave ISP and UART available for use, and I already had several on hand and was familiar with

them. Though the UART is not used in the final design, the serial interface was utilized extensively during prototyping, and may come in handy should you want to change patterns. Port C is interfaced directly to the inputs of the SSR board. In software, you can choose whether active high or active low is used to select a channel. I use the PIC16F690's internal oscillator since timing or high speed is not critical.

The SSR board shown in **Figure 2** is reasonably priced and less than the cost to buy the components to build it — let alone the cost to fabricate the printed circuit board (PCB). Each channel is optically isolated, fused, has an LED indicator that is useful when troubleshooting, and is built



■ **FIGURE 2.** View of the solid-state relay (SSR) board.



■ **FIGURE 1.** Eight-channel light controller schematic.

for operation at a household mains potential. With the SSR board operating at five volts and inputs spaced at 0.2", the interface to the microcontroller was quite easy. (Refer to the **sidebar**).

The maximum current for each channel is two amps, but with LED light strings you will be pushing much less current through each channel. In fact, during initial testing, I was concerned that one LED light string would not present a large enough load for the triac to fully turn on and off; fortunately, the switching worked well. I recommend using LED light strings because not only is their power consumption low, the colors that LEDs produce are more brilliant than incandescent lights shining through tinted glass.

Software

The software was written in C using Microchip's MLAB IDE (integrated development environment) and the HiTech PIC lite C compiler. The source code is available at the article link. Essentially, the program uses the random

number generator in C, *rand()*, to provide an ever-changing light show by randomly choosing from 15 different patterns and two speeds. The program is simple: First, the pattern and speed are chosen, then the pattern is executed a set number of times. It then loops back to pick a new pattern and speed, and executes the new pattern.

Generating Random Numbers in C

In C, *rand()* uses an algorithm to provide a series of pseudo random integers between 0 and *RAND_MAX* (a large number defined by the library). The algorithm is started by *rand()* and uses a seed generated by another function, *srand(int s)*. By using different values of 's' in *srand()*, the series of integers generated by *rand()* are different. Likewise, using the same values of 's' in *srand()* will produce the same series of integers.

You can see this for yourself by writing a simple program that prints out the sequence of integers produced by *rand()*. In this project, we ensure that the generated series is different each time the controller is powered up by adding one to the previously used seed which was stored in the microcontroller's EEPROM. The new seed is then stored in EEPROM so it can be incremented at the next power-up.

A quick Web search on *rand()* will bring up several discussions on the 'randomness' of it — especially when restricted in a small range as used in this program. If you are generating random numbers for encryption, unique identifications, or other security sensitive applications, this is a concern. In this application, by changing the seed each time the light controller is powered up, the randomness is good enough.

As noted earlier, *rand()* will generate a very wide range of integers from 0 to *RAND_MAX*. What if you only want a smaller range of integers (for example, 0 to 7) that is used extensively in this project? The modulus — the remainder after division — is used to restrict the range as shown in this code for the *RandomNumber* function:

```
char RandomNumber(unsigned char range)
{
    n = rand();
    char rnmbrr = n % range;
    return rnmbrr;
}
```

Here, the variable 'range' is (n+1) which restricts the generated random integers to between 0 and n inclusive. For example, for *range* = 8, the returned integer will always be between 0 and 7 since you cannot have a remainder greater than 7 when you divide by 8. For example, if the remainder is 8, then the number is evenly divided by 8 giving the modulus of 0. In the code, we start each cycle by choosing between two speeds by using *RandomNumber(2)* returning either a 0 or 1, and then choosing a pattern (or

The Tree

The tree that is used with this controller has a center pole from which eight LED light strings are hung, flaring out to form a tree-like structure. Each LED light string makes two legs, giving a total of 16 legs. A pleasing proportion is for the height of the center pole to be twice the diameter of the base of the tree. I used an eight foot pole with a base diameter of four feet. The middle of each light string is at the top of the pole, with the ends at the base. Each light string is plugged into the controller sequentially. The controller sits on the base of the tree.

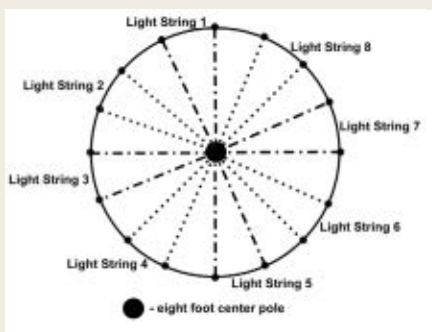


FIGURE A. Drawing of the tree as viewed from the top. Eight sets of LED lights result in 16 legs.

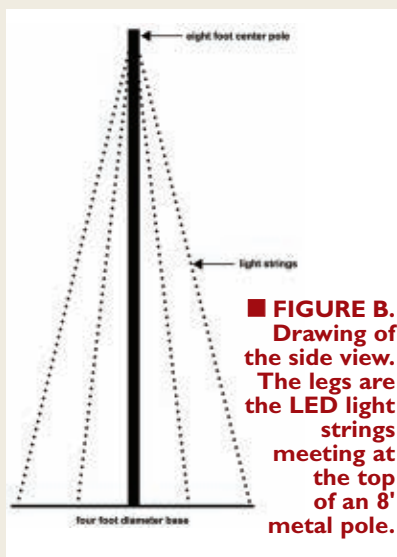


FIGURE B. Drawing of the side view. The legs are the LED light strings meeting at the top of an 8' metal pole.

mode) by using *RandomNumber(15)* and returning an integer between 0 and 14.

You may notice there is a test to see if the pattern (mode) would repeat. I found in my early prototypes that patterns and the actual display would repeat periodically. This occurs because we are restricting the range of our random number generator. Even though the random number generator produces two different integers, they will appear to repeat if they have the same remainder. I found that generating another random integer will cut the repeats so they are infrequent and not obvious to the observer. I used the same test in several of the patterns, too.

Using Port C to Drive SSR Board Channels

Another one of the reasons I chose the PIC16F690 was that port C is eight bits wide, so it is simple to write a value to port C. Writing to the *PORTC* register allows you to directly write the desired state for each output. Bit 0 of *PORTC* is the value for C0; bit 1 is the value for C1 — all the way to C7. For example, *PORTC* = 0b10101010 will cause all even outputs to be low and all odd outputs to be high. Once the pattern is selected, the corresponding routine is executed to turn on the desired channels of the SSR board. Let's take a look at the function that drives the corresponding port C outputs of the microcontroller:

```
char PortCDriver(char selected, char nchan, char
onchan, char logic)
{
    space = 0;
    if (nchan == 2){space = 4;}
    if (nchan == 3){space = 3;}
    char port=0xFF&0x01<<(selected |
0xFF&0x01<<((selected+space)%8) |
0xFF&0x01<<((selected+space*2)%8));
    char flip=onchan^logic;
    //determine if outputs need to be toggled
    if (flip){port=port^0xFF;}
    PORTC=port;
    return port;
}
```

There are a number of parameters for this function: *selected* is the active channel for the pattern; *nchan* is the total number of channels to be active (1, 2, or 3); *onchan* sets whether the active channels are lit or dark; *logic* is positive or negative logic.

With this function, we can take the channel that has been selected by the pattern's routine and add an additional one or two channels equally spaced around the perimeter of the tree. This will give an appearance of channels dancing around the tree. Notice that the spacing is added to the selected channel and modulus 8 is used to keep the result between 0 and 7.

For example, if the pattern calls for three active channels and the selected channel is 2, the other two active channels will be 5 and 0 (i.e., $(2+2*3)\%8 = 8\%8=0$).

These channels are ORed together and written into *PORTC* after it is determined if the bits need to be toggled — which depends on whether the active channels are lit or dark and which logic is required. By understanding how each parameter affects the operation of this function, you will be able to build your own set of patterns.

Examining a Few Patterns

To get some insight on how the display patterns are generated, let's take a look at two patterns: one that is random, and one that is sequential. In operation, the particular pattern and the speed is chosen using random number generators at the beginning of each *while(1)* cycle.

If pattern (i.e., mode) 4 is selected, then the following code — using yet another random number generator — is executed. This pattern specifically blanks the active channel, as well as an additional channel spaced by four channels; the other six channels will be lit. As you will see, this pattern gives the appearance of two dark strings directly opposite each other dancing around the tree:

```
// Blanks only channel corresponding to
generated number plus one spaced channel
// For two enabled channels the spacing is four,
for three enabled channels the spacing is three

if (mode==4)
{
    for (i=1; i<=reps; i++)
    {
        char eight = RandomNumber(8);
```

Why use solid-state relays instead of electromechanical relays?

The easy answer is solid-state relays (SSR) don't make any noise. Plus, they are much more reliable than electromechanical (EM) relays. In fact, an EM relay would likely not survive a holiday season of operation! The lifetime for EM relays is specified in the number of contacts made — about 100k contacts for a typical EM relay. Because there are no moving parts, the lifetime for an SSR is specified in hours of operation; minimally 100k hours, though many SSRs have lifetimes in excess of one million hours.

Let's assume that the light controller is used between Thanksgiving and New Year's Day (35 days) for six hours per day. If we assume a conservative one contact per second, the number of contacts made would be 35 days * 6 hours/day * 3600 seconds/hour = 756k contacts. This is well over the predicted lifetime of 100k contacts. Meanwhile, the SSR is operating for 210 hours which is a small fraction of the 100k hours lifetime.

Since we are switching very low currents with the EM relay, in the case of LED light strings, we can generously increase its lifetime to one million contacts. Even then, the story is still not too promising. Under those conditions, an individual EM relay's probability of surviving the holiday season is only 47%, while for the SSR it is 99.8%. For eight relays, the probabilities of survival for the system using EM relays or SSRs are 0.24% and 98.4%, respectively, using the very low current assumption.


```

        if(eight%4==oldeight%4)
        {
            eight = RandomNumber(8);
        } //end if
        oldeight=eight;
        PortCDriver(eight,2,0,portclogic);
        pause(dd);
    } // end for
    } // end if

```

The pattern is executed 32 times as defined by the variable *reps*. During each execution, a random integer between 0 and 7 is generated and is used to select the active channel. Once again, we test for repeats as discussed earlier, but this time we use modulus 4 instead of modulus 8. This is because we will be activating two channels, with the second channel spaced four channels from the selected channel.

For example, selecting channel 0 looks the same as if channel 4 was selected. Once the active channel has been selected, the *PortCDriver()* function is used to drive the SSR board. Here, we designate the selected channel with the variable *eight*; there are two active channels, the active channels are blanked, and the port logic is defined by the *portclogic* variable. *Pause(dd)* freezes the pattern for either 250 milliseconds (*dd* = 0) or 500 milliseconds (*dd* = 1), setting the speed of the display. In this next example, two opposite channels are active and will be lit while the other six sets of light strings will be dark. The pattern will appear to rotate around the tree four times. This pattern is executed when pattern 14 is selected:

```

// Enables only one channel and spaced channel
// decreasing rotation

    if (mode==14)
    {
        for (i=reps; i>=1; i-)
        {
            char eight = i%8;
            PortCDriver(eight,2,1,portclogic);

            pause(dd);
        } // end for
    } // end if

```

Similar to the prior example, this pattern is executed 32 times, but in this case the selected channel is not random, but rather the modulus 8 of the rep count as it decreases from 32 to 1. Since I chose 32 for the number of repetitions of patterns, the cycle will repeat four times. Again, the function *PortCDriver()* is used to drive the SSR board with the selected channel designated by the variable *eight*; the number of active channels is two, the active channels will be lit (i.e. enabled), and the port logic is defined by *portclogic*.

Construction

This project involves switching the mains voltage and currents that can be dangerous to human life. Your safety

and the safety of others should be the first priority as you build this project. If you are not experienced or don't feel confident in working with mains voltages and currents, get help from someone who is experienced in dealing with household voltages and currents.

Enclosure

Since my controller is outside at the base of the tree, a top concern for me was to keep water and prying fingers from entering the enclosure. To do this, I minimized the number of holes that could let water in by eliminating any adjustments or other controls. My controller — and the rest of my holiday lighting — is powered by an outlet connected to my home automation system, so I left out the power switch. The only holes needed are for the power in and each of the switched channels. In my case, I used a plastic food storage container that is waterproof. It worked well for me since it is sealed against the elements and is electrically isolated. It is not secure enough to prevent access while it is powered up, however. If contact with the box is possible, then you should use a waterproof enclosure that is secured with fasteners or move the controller into a secured location.

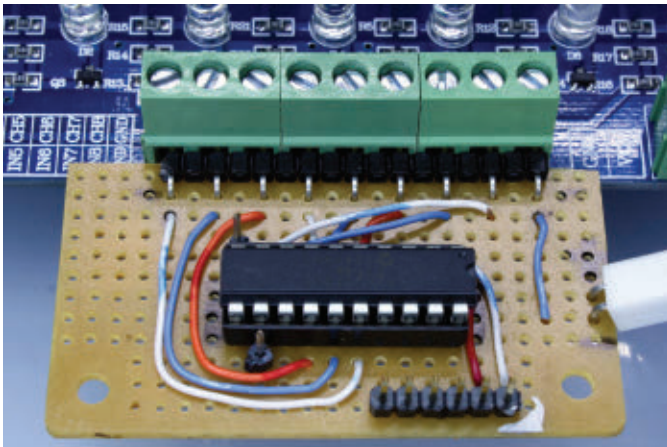
To bring power to the controller and deliver switched power from each channel to the light strings, I used polarized power extension cords with the unused end cut off. These cords have polarized plugs and outlets with one large blade (neutral) and one small blade (hot). During construction, keep proper polarization so that you are fusing and switching the hot side, while neutral is only used as the return. After cutting off one end of the extension cord, you can use an ohmmeter to identify which lead is hot. This is the side that will be switched by the SSR board.

To keep the enclosure waterproof, I had to make sure each power cord entry was sealed. I used a grommet to protect the cord from the plastic edge of the hole it passed through. Since the cords that I used were flat rather than round, fitting the grommet was more challenging than simply drilling a round hole. I used two smaller adjacent holes to form an elongated hole. After clipping out the plastic between the two drilled holes, an elongated hole just larger than the cord was produced.

The cord is pulled through the hole after fitting the grommet. Threading the cord through the grommet should be tight; I practiced this on a scrap piece of plastic first to get the right size and fit. In addition, I used a plastic zip tie around the cord on each side of the hole to keep the cord from moving, then some silicone as a final water tight seal.

Microcontroller Board

Since the circuit is simple, I used a perf board for the microcontroller related circuitry. The layout of the board



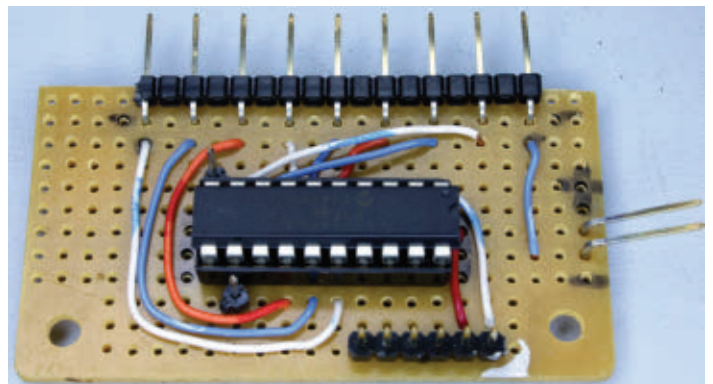
■ **FIGURE 3. Microcontroller board mated to the SSR board.**

was arranged so it would easily interface to the SSR board. The input terminals of the SSR board are spaced 0.2" for all eight channels and ground. I used 17 positions of a right angle male 0.1" header; keeping the outer pins, I removed every other pin so nine 0.2" spaced pins were left. I soldered the shorter pins on the edge row of the perf board, with the longer pins extending out from the board so they could mate with the SSR board. This takes care of the connections for channels 1 through 8 and the ground connection. Next, wire the header pins to the appropriate pins on the microcontroller. The completed microcontroller board is shown mated to the SSR board in **Figure 3**.

Looking at the microcontroller board in **Figure 4**, you will notice pins for ICSP, Rx, and Tx for future use or for modifying the program; these pins are optional. I used a socket for the microcontroller with the decoupling capacitor, C1, mounted between the two rows of the socket beneath the socketed microcontroller. This allows the capacitor to be very close to the Vcc and ground leads while taking up less space. The only other connection is a male two-pin header for power, fed from a five volt wall wart supply.

SSR Board

Using the prebuilt SSR board makes wiring the high voltage mains circuitry easier and safer. Even so, precautions must be taken to make sure polarity is maintained; there is no exposed high voltage and use of wire size that can handle the current that will be passing through it. As shown in **Figure 5**, leads are dressed and color-coded so polarity is maintained; the hot side of the AC should be switched by the SSR board while neutral is used for return (or common). Traditionally, hot is connected with black insulated wire and neutral with white insulated wire as used here. Be sure that the wire insulation is rated for household voltage.

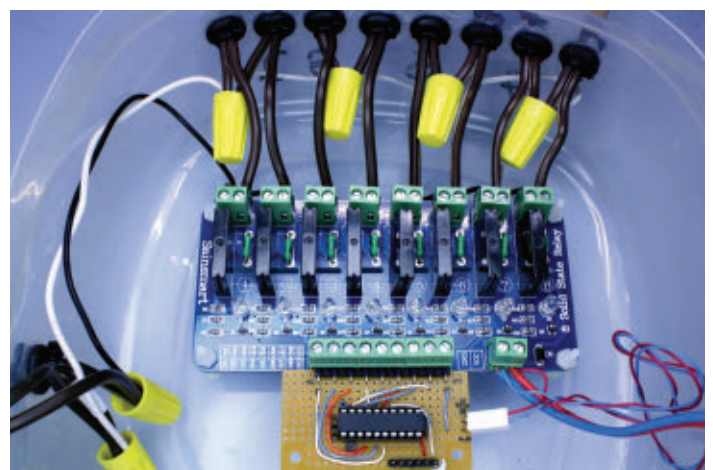


■ **FIGURE 4. Microcontroller circuitry built on a perf board.**

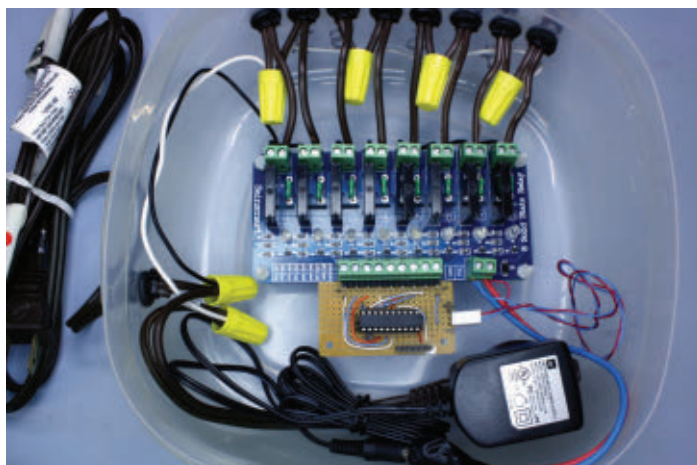
Each channel of the SSR board is rated for two amps, so you should limit current draw to less than that. In my case, I was driving a single string of LED lights that draws very little current. Depending on your current draw, make sure your wire size is sufficient; for two amps, #18 wire will be adequate. For the common hot and neutral wires feeding the SSR board and lights, the current draw will be eight times the individual channel current draw. In my case, the total current draw is still low, but could be significant in your configuration. If each channel is passing close to its two amp limit, your common feeds need to be at least #14 wire. If you don't have experience in determining current draw and sizing wires appropriately, get help from someone who knows how to do this.

Connect the hot side of the outlets for your lights to the switched side of each channel on the SSR board. The neutral side is shared among all channels; use wire nuts to connect the neutrals together. Don't do this with just one wire nut. Connect two channels together to the common neutral with one wire nut and daisy-chain to the adjacent channels. This not only gives a neater layout but also puts less stress on the connections.

Plastic screws, nuts, and standoffs are used to support



■ **FIGURE 5. Close-up view of the AC wiring.**



■ **FIGURE 6. Completed assembly before installing the cover.**

the SSR board. The SSR board and the connected perf board float in the case, held in place by the connecting wires. This was done to eliminate additional holes that would be used to mount to the case, reducing possible compromise of the weather seal. The power to the SSR board is also supplied by the five volt wall wart used to power the microcontroller board; connect power to the Vcc and GND terminals on the SSR board. The completed assembly is shown in **Figure 6**.

Startup and Troubleshooting

After building the microcontroller board, it is easy to check for operation. Connect the microcontroller board to the SSR board and supply power from the wall wart to both boards; no need to hook up the outlets or light strings just yet. The SSR board called out for this project has an LED on each channel that lights when the channel is selected. When powered up, you should see a series of changing patterns of lit LEDs on the SSR board. If not, make sure that you have properly connected the microcontroller board, and check your perf board wiring.

Once this test has passed, go ahead and place the boards into the enclosure and attach the power and switched outlets for the light strings.

There are no adjustments needed — just supply power and the controller will run on its own.

Make sure the voltage to the SSR board is close to five volts. I found that if the voltage is low, the selected channel LED will still light but the solid-state relay may not turn on. This confounded me for a while during prototyping until I found that my voltage had dipped below five volts.

Modifications You Can Make

There are modifications to the controller you can make including patterns, speed, and the number of channels. You can easily add or modify the patterns that the controller generates. When I was prototyping the circuit, I drove the LEDs directly with the appropriate series resistor from the PIC16F690. I arranged the LEDs in approximately the same pattern as the final display which, in my case, was a circle. This gave me an idea what the pattern would look like, which I found quite useful.

If you add or delete patterns (modes), be sure to change the modulus where the mode is selected. You can change the speed of the pattern by modifying the delay in the *pause()* function, and the number of times the display is stepped by modifying the variable *reps*.

Modifying the number of channels is more complicated, but can be done by experienced builders. The *PortCDriver()* function may need to be modified, and the range of random numbers generated in patterns would have to be modified, as well as the modulus used in other patterns. If you want to use more than eight channels, then other ports of the microcontroller will have to be used, as well as additional SSR boards.

I hope you will enjoy building this project. It made a big difference in my holiday display. It will do the same for you. **NV**

ITEM	DESCRIPTION	PART #	NOTES
BRD1	8 channel 5V SSR	20-018-902	www.SainSmart.com , eBay, or Amazon www.sainsmart.com/8-channel-5v-solid-state-relay-module-omron-ssr-4-pic-arm-avr-dsp-arduino.html
C1	0.1 μ F		
F1	Fuse and fuse holder		Sized per total current demand
J1a	17 pin .1" male right angle header		Mouser, Digi-Key, Element14
J2	Six-pin .1" male header		Mouser, Digi-Key, Element14
PS1	5 volt \geq 200 mA wall wart		
U1	Eight-bit PIC microcontroller	PIC16F690-I/P	Mouser, Digi-Key, Element14
Misc	Polarized extension cords		Hardware store
Misc	Wire nuts		Hardware store
Misc	Zip ties		Hardware store
Misc	Silicone		Hardware store

PARTS LIST

Boxed Kit Amps

Gobo Stereo Amplifier Build

A fun to build desktop stereo amplifier kit for any experience level.

I've been looking for a small high quality stereo amplifier to use on my test bench for a while, and happened across Boxed Kit Amps and their Gobo stereo audio amplifier kit. They sent one for me to build and evaluate, and I have to say that I'm very impressed. The kit was both easy and straightforward to build, and it was fun to see it taking shape as each board was finished. What really sets it apart from other amp kits out there is that it includes 100% of the parts needed to complete the project — including the power supply and the case — and it includes top quality passive components.

This amp kit is based around the common LM1875 chip amp. It can reportedly deliver up to 18W audio output with as low as 0.015% total harmonic distortion, and it's wrapped in a beautiful and very interesting looking laser cut blue acrylic case to show off the construction.

Unboxing

The packaging is very secure, and thick foam end pieces and separate compartments for the different bags of parts keep the acrylic safe in transit. The parts were grouped based on when you'd need them in the instructions — the amplifier modules, power supply module, interconnect and chassis hardware, and the transformer were all packed separately.

The first thing I noticed was the quality of the components included. Boxed Kit Amps used high quality audio-grade parts in the kit, including Nichicon electrolytic capacitors, and WIMA and Panasonic film capacitors on the amp modules. I really appreciated this attention to detail. A few key wiring diagrams are included in the box, with the full build manual provided as a PDF on Boxed Kit Amps' website at www.boxedkitamps.com.

It's about 50 pages long, so I can see why they didn't include it as a print-out. I ended up printing off a copy to have on my bench as I went through the

Figure 1. Finished product.

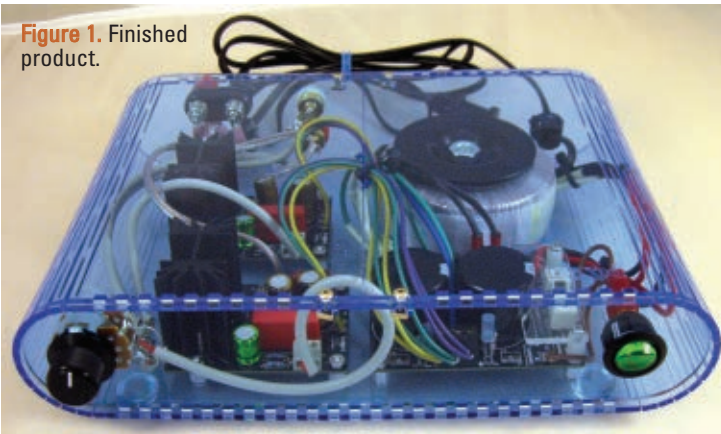




Figure 2. Everything held safely in place.



Figure 3. Parts are well organized by function and step.

process, but it'd be easy enough to have it up on a laptop or tablet nearby to save a few trees.

I followed the build instructions as written to get the full experience of the kit. First up were the amp modules.

Amplifier Modules

There are two identical amplifier modules in the kit which stay pretty close to the reference design for the LM1875. They have a very simple signal path — there's just one capacitor and the volume control between the input jacks and the input to the amp chip.

The build instructions are easy to follow, and the boards themselves are great quality. Each PCB (printed circuit board) has an eye-catching black solder mask which is a lot more interesting to look at than the generic green style you usually see. The board uses entirely plated through-holes which form a strong physical and electrical connection.

Populating the passive components was easy, but it was a little finicky getting the amp chip mounted to the heatsink. Nothing a pair of tweezers and a steady hand couldn't deal with, though. The solder pads on the

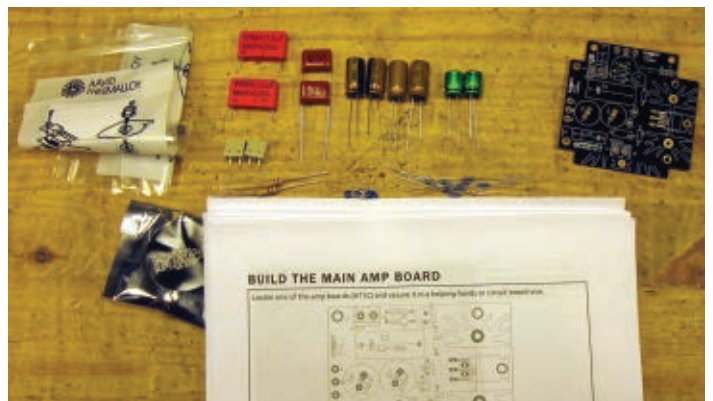


Figure 4. Amplifier components.

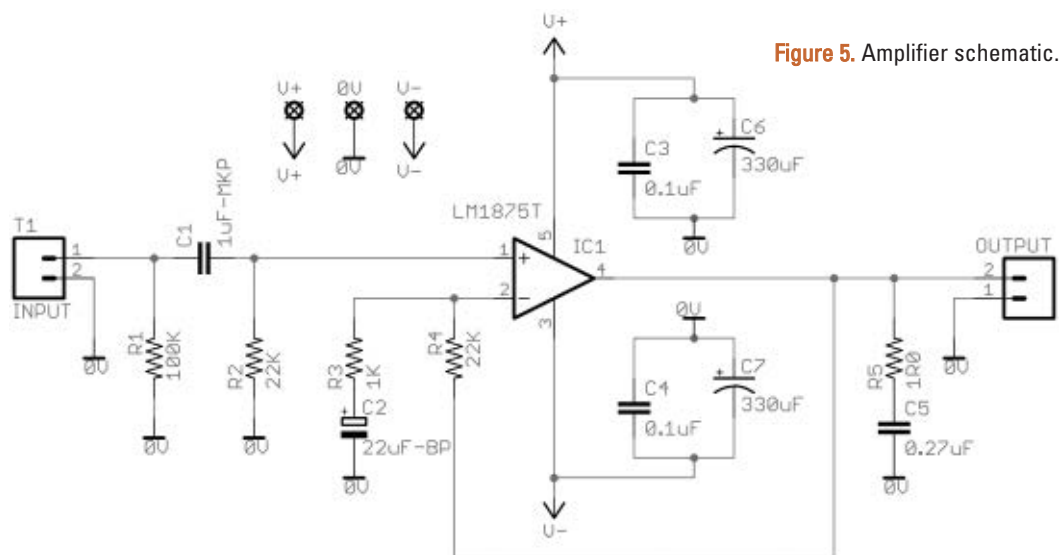


Figure 5. Amplifier schematic.



Figure 6. Completed amplifier board ready to mount the LM1875.



Figure 7. Securing the silicone thermal pad.



Figure 8. Chip mounting complete.

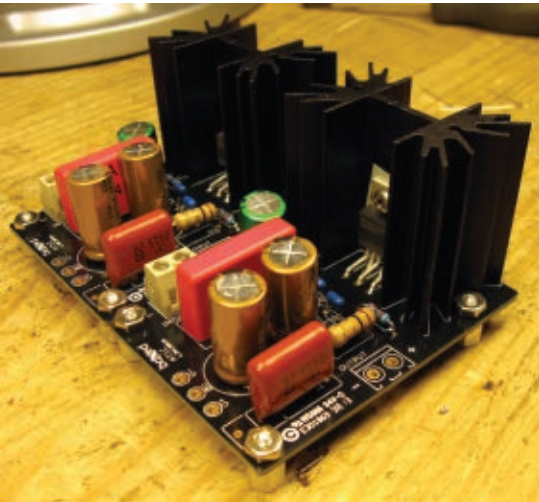


Figure 9. Both amplifier modules side by side.



Figure 10. Power supply components.

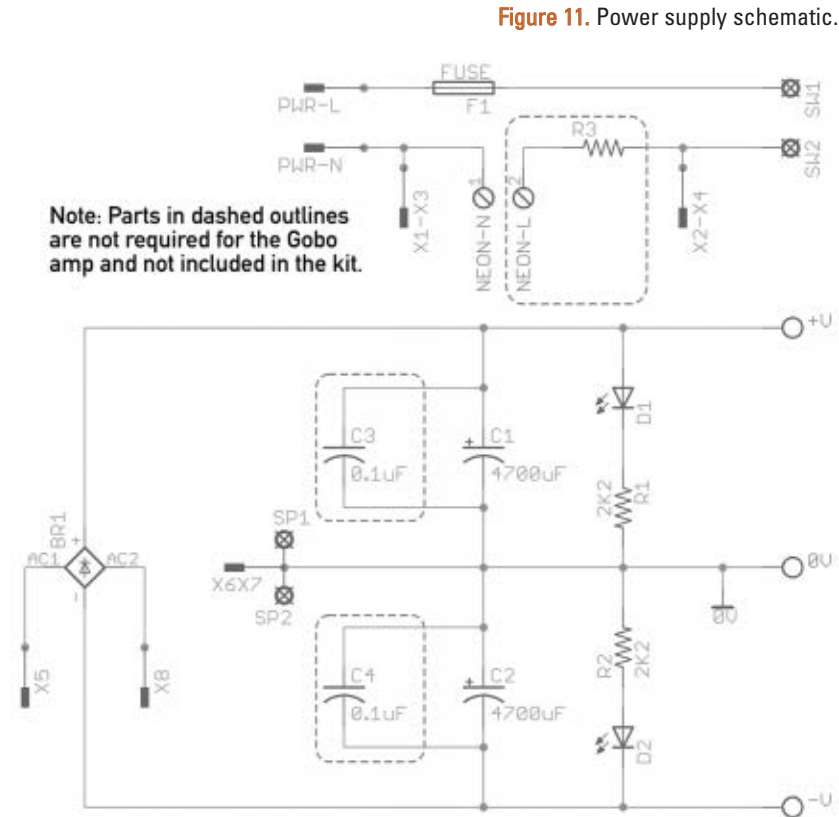




Figure 12. Mounting the bridge rectifier, terminals, and LEDs.

bottom were a bit narrow, but were easy enough to work with using a fine point on the soldering iron.

With both amplifier modules built, it was time to move on to the power supply.

Power Supply

The power supply board is shared with a few other Boxed Kit Amps projects, so has some extra pads that aren't meant to be populated in this version of the kit. It's a straightforward power supply — a toroidal transformer feeding a bridge rectifier, filtered by 4,700 μF capacitors to form a bipolar power supply. One nice touch are the LEDs across the filter caps as bleeders. They could be replaced by a resistor if you don't want another light source in the case, however.

First power-up and no smoke! The LEDs are cool sitting there on, and the power switch lights up when turned on, too. A quick check of the voltages shows they're within spec. The power supply might benefit from a little better voltage regulation — the positive and

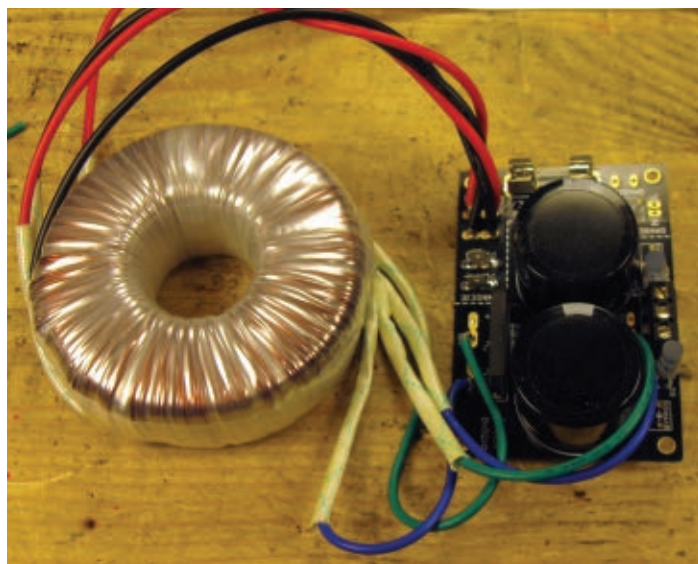


Figure 13. Adding the transformer.

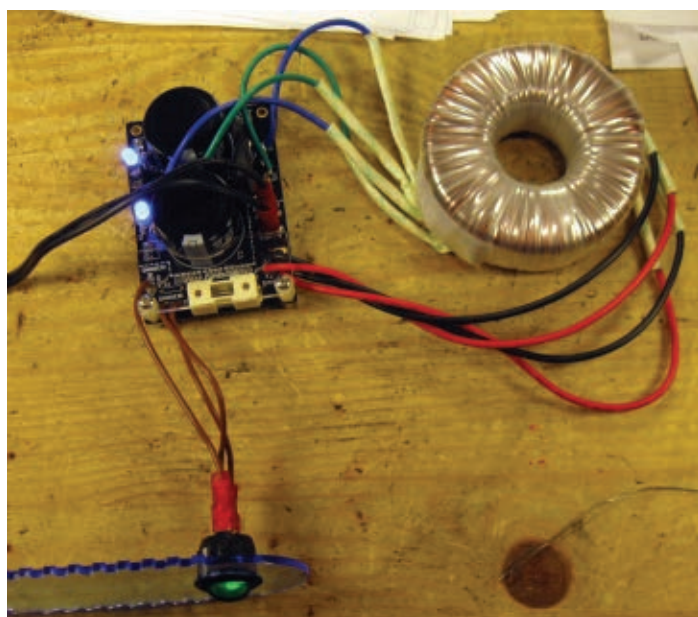


Figure 14. First power supply test.



Figure 15. Measuring the unloaded output voltage.

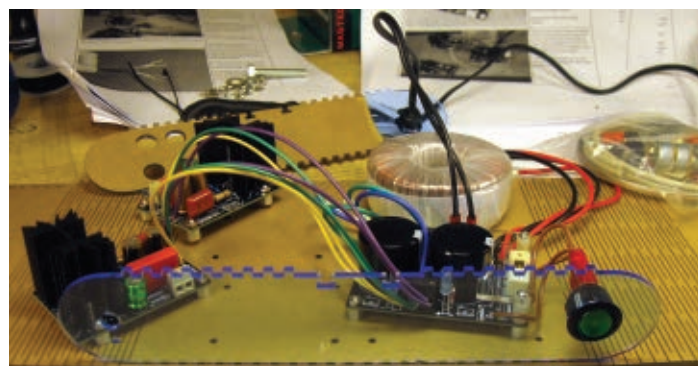


Figure 16. Laying out the components.

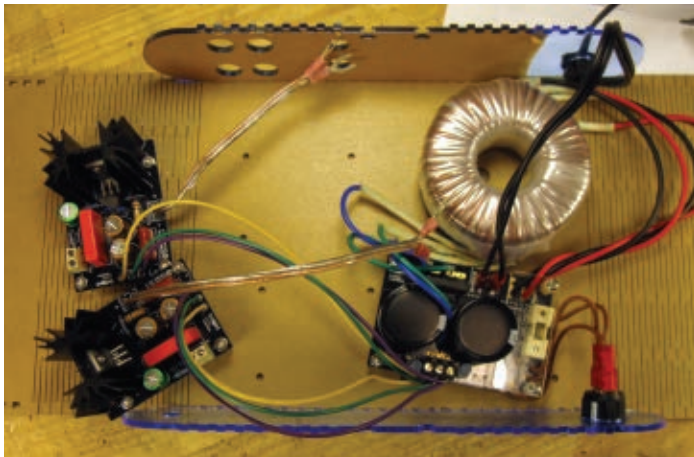


Figure 17. Mounting the hardware.

negative rails are close, but are not quite perfectly matched.

Finishing Up

Finally, it was time to hook everything together. I started staging the boards roughly where they mount inside the enclosure and hooking it all together. Wiring the boards together was a bit tedious, but uneventful. I did find the holes for the wires to be a bit small – especially the speaker output leads. It took several tries to remove the right number of individual strands from the bundle before they seated nicely, but otherwise it wasn't too bad.

On the power supply board, a screw-down terminal block holds the DC power wires. On the amp modules, those wires are soldered in. Audio input is a screw-down terminal, and speaker outputs are soldered in place.

By now, you're thinking "all this plastic, it's not shielded at all – how is it going to protect against noise and interference?" And, you'd be right. It is all plastic, but luckily interference and noise aren't an issue as the signal path is fully shielded – from the input jacks through coax

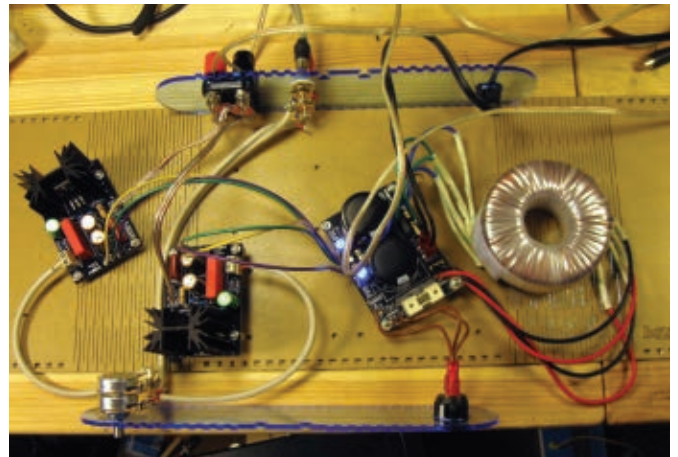


Figure 18. First full system test.

to the volume control with a grounded metal body, and through more shielded coax to the amplifier input.

Luckily, there was no smoke on the fully assembled test run, either. I did do a couple of tests. First, with no input signal, I turned the volume to max and listened for white noise in the speakers. It was nearly completely silent – very nicely done. Then, I hooked it up to my meters. I got about 12W with low to virtually no distortion into an 8 ohm load (less than 0.06% THD), 15W with moderate distortion (<2% THD), and past 15W up through 25W measured power with distortion starting about 30%.

With confirmation it was working well, I zip-tied and bolted everything together. Some of my wire routing was a bit longer than I liked, but with the fully shielded audio path, I wasn't too worried about an extra inch or so in a couple of places.

The case itself is beautiful. It's a laser cut piece of transparent blue acrylic with a very interesting etching pattern to let it wrap around and form rounded corners. The entire main section of the case is made of a single piece of acrylic, with separate front and rear panels with laser cut holes for the inputs and outputs. It came with a

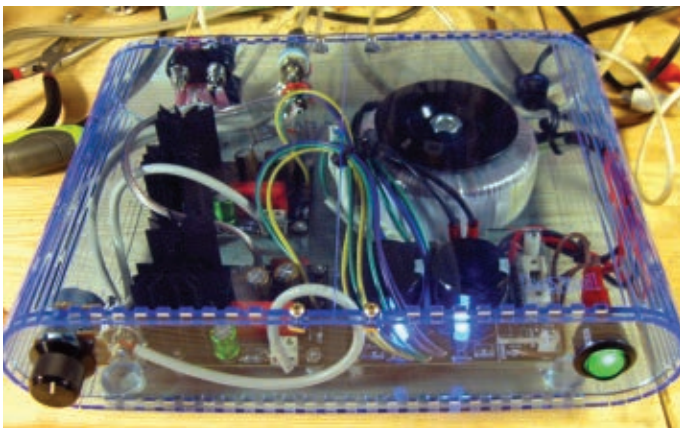


Figure 19. Completed amplifier, front.

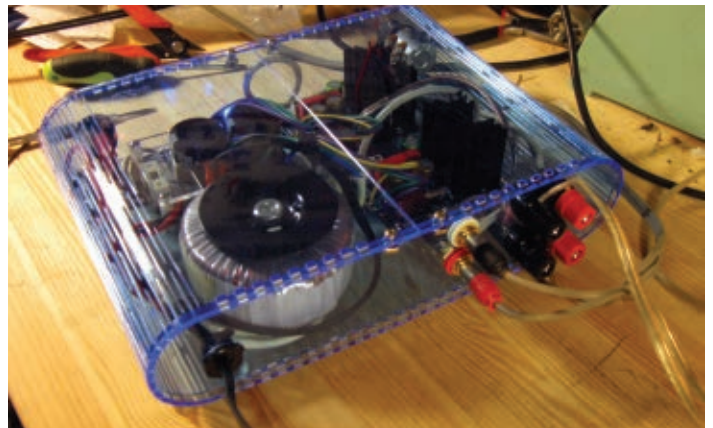


Figure 20. Completed amplifier, rear.

protective film on both sides.

Removing the film was probably the most annoying part of building the kit — it was easy on the flat surfaces, but the curved portions were a real challenge due to the cuts. It probably took me 20 minutes of picking little bits of the protective wrapper off to get the acrylic panels ready to finally assemble.

After I got the film off, the residual static charge it left on the acrylic was enough to attract every speck of dust from around the room onto its surface, so I wiped it down with a clean (just barely) damp cloth and that seemed to take care of it.

Getting the front and rear panels to mate neatly was a bit challenging too. It felt like it required three hands to snap everything together. However, I did get all the tabs lined up correctly in the end, and bolted them down with the small securing screws.

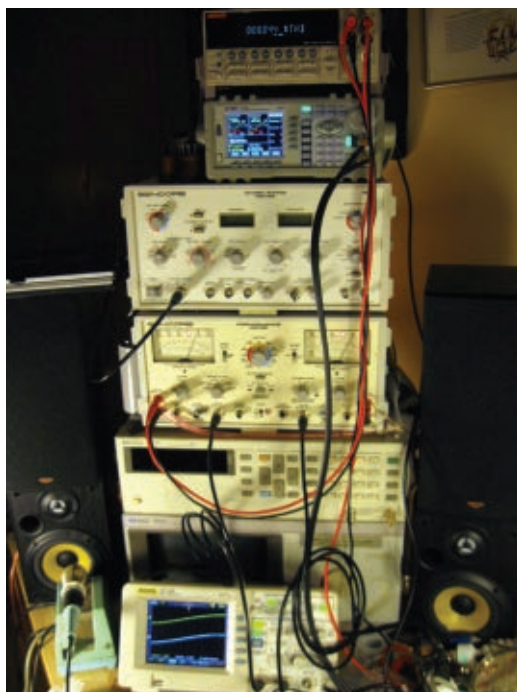


Figure 21. Test stack.

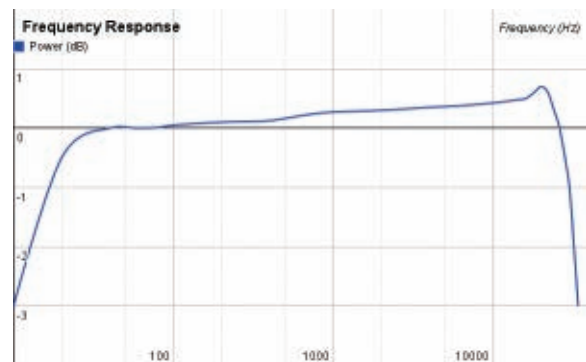


Figure 22. Frequency response (0 dB = 12W into 8 ohms).

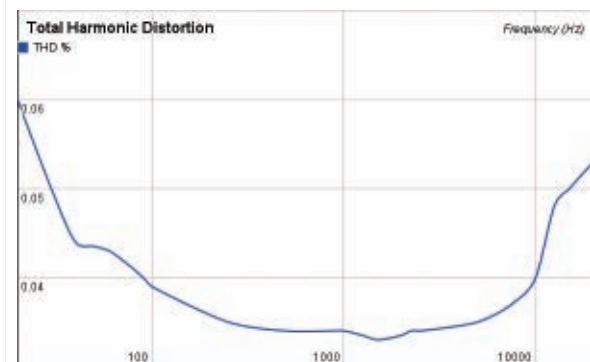


Figure 23. Total harmonic distortion (N = 5 at 12W into 8 ohms).

The lead dress isn't perfect inside, but it'll be easy to go back in and clean it up after the fact. I did a few listening tests with it finally assembled. This amp doesn't put out a huge amount of power, so it's best with high efficiency speakers. I used a set of Klipsch bookshelf speakers with about 94 dB efficiency, and was able to drive them louder than I would want to listen to at my desk. So, there's plenty of power available.

As far as the sound, it was accurate and neutral, almost clinical even. The limiting factor of the sound quality seemed to be the source material. Since I'll be using it largely as a desktop amplifier, I used a selection of lossless digital audio files and a FiiO external DAC. I could hear every note and pick out every instrument with perfect clarity. It did manage to sound

decent with a Pandora stream, but it wasn't hard to pick out some compression artifacts which the amp faithfully reproduced if you listened for it. It's going to sound as good as the signal you feed it.

After the listening test, I took some measurements. Based on my analysis, I found the amp's power bandwidth was within 1 dB from 20 kHz; 33 kHz and down was only -3 dB at 10 Hz. Distortion was very low throughout the frequency range, too. It was a little higher at the ends and lower in the middle, but all below 0.06% THD. That's a little bit higher than the 0.015% theoretical performance of the chip; I expect I'm a bit limited by my test setup in this case. Regardless, total harmonic distortion below 0.06% is very, very good.

It took me about six hours from start to finish to build the kit, and other than some trouble prepping the case and attaching some of the stranded wires, it was fun and straightforward. If you're a new builder, it might take a little longer.

The modular design and well thought-out instructions make each step a rewarding experience. Since the kit is modular, it's easy to make some progress even if you only have a few minutes to dedicate to projects in an evening.

I'd recommend this kit if you're looking for a fun project that delivers great results. I'm planning to use the completed amplifier as my bench amp to use while working on line-level audio devices like tuners and pre-amps since the low distortion will ensure I'm hearing the device under test, and not the amp itself. It'd be just as great as a desktop amp for daily listening, too. And, it looks great! **NV**

The Arduino Classroom

Arduino 101 — Chapter 11: Sound

Doing Things in the Background: Interrupts

Microcontrollers — like the Atmel ATmega328 used in the Arduino UNO — are made of several internal sections on an **IC (Integrated Circuit)** that work together to provide intelligence, communications, sensing, and control. The intelligence is provided by the **CPU (Central Processing Unit)**: a section that runs (processes) the code you write one step at a time.

Peripheral devices do the communications, sensing, and control. These are sections on the IC that can do things independent of the CPU. For example, the Arduino serial communications use a **UART** peripheral device to communicate with a PC. The CPU loads a byte of data into the UART and tells it to send the byte. The UART then — with no further help from the CPU — sends each of the eight bits of the byte, one at a time to the PC.

The UART also receives data from the PC; again, one bit at a time until it has the required eight bits for a byte of data. This sending and receiving of bytes of data happen at the same time that the CPU is processing unrelated commands elsewhere on the IC. This work is done as **background processing**, meaning without requiring the CPU to do the work.

Another peripheral device we have used is a timer for analog voltage output. We — via the CPU — use the *analogOutput()* function to tell this timer peripheral device to create a PWM (Pulse Width Modulation) signal and output it on a specific pin. The timer then generates the PWM signal on the pin — also in the background without further use of the CPU.

Likewise, we measure voltage on some pins by using the *analogInput()* function that reads the ADC (Analog-to-Digital Conversion) peripheral.

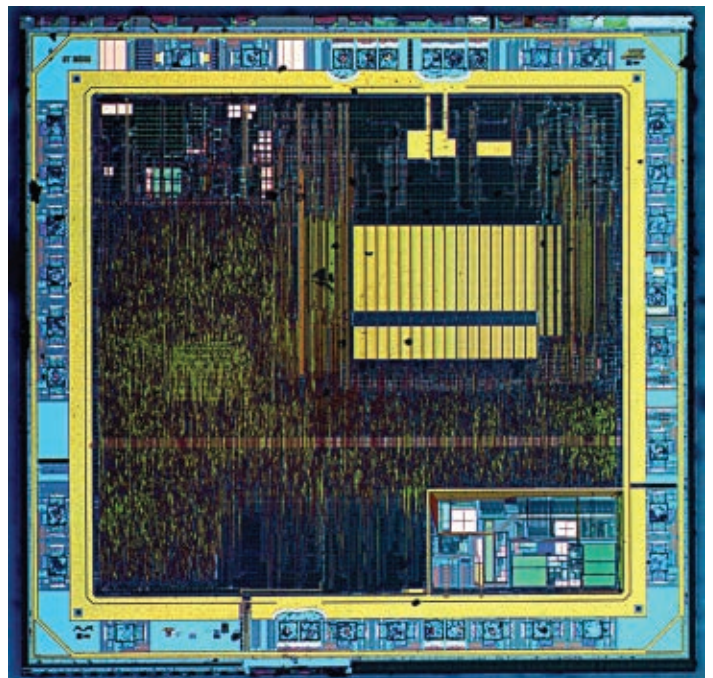
In early designs of microcontrollers, some of these sorts of tasks were done directly by the CPU, but they take up a lot of CPU time. Later designs added the peripheral devices so they could do common repetitive tasks without burdening the CPU.

To make an analogy, you can think of a microcontroller as a small company with a boss and some

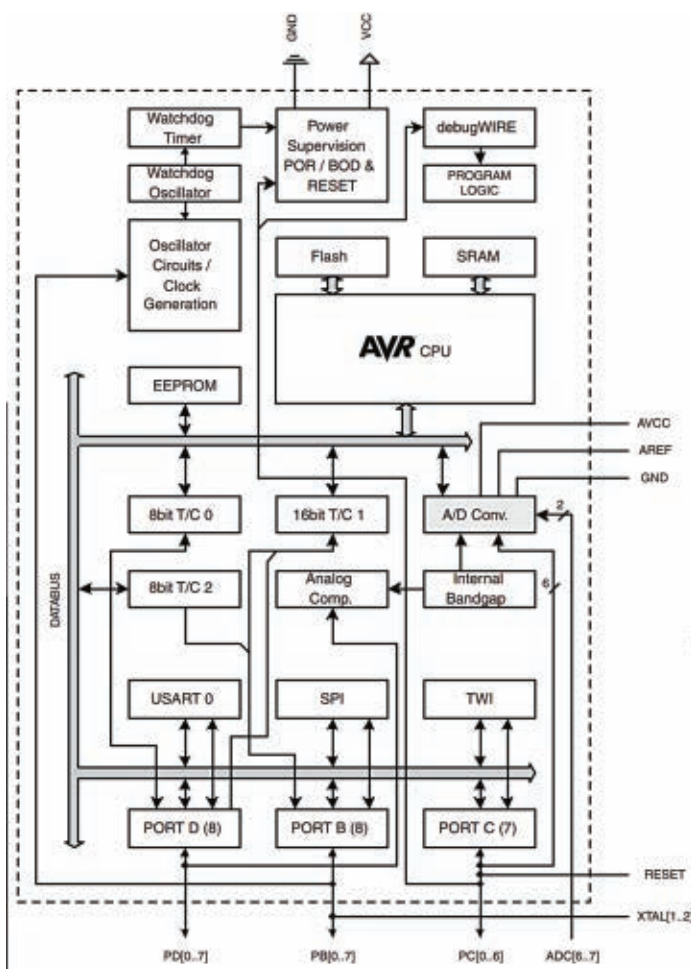
In this chapter, we will learn to make sounds using the Arduino. In order to make sounds, we will need to learn about arrays and interrupts.

employees. The boss is up front and in charge, while employees are laboring away in the background. Think of the CPU as the boss, and the program you write as the tasks that need to be done. The boss may do some of those tasks, or may assign other tasks to employees (who are the equivalent of the microcontroller peripheral devices).

These separate sections are visible on the silicon in the IC as shown for an Atmel chip in **Figure 1**. These



■ **FIGURE 1: ATmega AVR silicon.**



■ FIGURE 2: ATmega328p block diagram.

sections correspond to those shown in the block diagram for an Arduino UNO's ATmega328p in **Figure 2**.

Much of what you see in these illustrations is well beyond the scope of this series, but they help understand the general concept that there are separate entities on the silicon that surround and work with the CPU.

So, in this boss/employee analogy, how does the boss find out about the status of the employee's work? There are two ways to do this. One is for the boss to stop working and go see what the employees are doing. In the computer world, when the CPU (boss) checks on the peripherals (employee), it is called **polling**.

The other way to do this is for the employees to knock on the boss' door and interrupt his workflow; for computers this is called an **interrupt**. The boss responds to the racket by setting the current work aside — perhaps marking the task list so the work can be resumed when the interruption is taken care of — then, the boss deals with the interruption. Once it's dealt with, the boss looks back at the task list and resumes where the mark he left shows the interruption occurred. For a computer interrupt, the CPU

marks where it was in the main code, deals with the interrupt, and then returns to the code where it left off.

There are several types of interrupts on the Arduino. One is a **hardware interrupt** that can notify the CPU when a voltage changes on a pin. Another is a **timer interrupt** that tells the CPU when a time interval has passed.

A hardware interrupt can be generated by a voltage change on either Arduino pin 2 or 3. We will use this to detect when our light sensor indicates that — after being dark— it suddenly becomes light again. Imagine a design that is used only when the light comes on in a room.

The Arduino could sit there and continuously check the light sensor (polling), or the light sensor could be connected to a pin with hardware interrupt capability. This way, the Arduino only has to deal with the light change when a light change actually happens and sends an interrupt to the CPU. We will use the Arduino *attachInterrupt* library to do this in Lab 1.

A timer interrupt is generated when a timer reaches a certain preset value. For example, you might program a timer to interrupt the CPU after one second has passed. We will use the Arduino *TimerOne* library to do this in Lab 2. In later labs, we will use this kind of interrupt to generate sound in the background while the CPU does other things.

Making Sounds

A musical tone is characterized by its pitch, duration, intensity, and timbre. Pitch is a human perception based on the frequency of a sound. Duration at its simplest is the length of time for a musical note. Intensity is the loudness. Timbre refers to perceived qualities of the tone that differentiate between various musical sources. For instance, a given pitch may be produced by both a violin and a trumpet but the timbre is different.

Our piezo speaker has many problems associated with creating a musical tone. The intensity of each pitch varies significantly; far greater intensity for pitches are near the resonant frequency (frequency that is physically amplified by the box containing the piezo element) and then are very diminished elsewhere. We do have decent control over the duration, but the timbre can only be described as awful — certainly not Hi-Fi (**H**igh **F**idelity = really good sound).

Nonetheless, we can use a piezo element to produce something that people will generally recognize as music — albeit awful sounding music, but still recognizable.

For our purposes here, music theory even in its most elementary form is beyond the scope of what we are trying to accomplish. We will thus just say that music involves tones played for discrete periods of time.

An example of this would be the tune, Happy Birthday which can be played with 26 notes — each held

for a relative duration of one, two, four, or six equal periods:

Happy Birthday

Notes: ccdcfeccdcgfccCafedbbafgf

Duration for each note: 1, 1, 2, 2, 2, 4, 1, 1, 2, 2, 2, 2, 6, 1, 1, 2, 2, 2, 2, 4

Try singing the song while looking at the duration for each note; this should make more sense. The first six durations are 1, 1, 2, 2, 2, 4, which correspond to "Hap py birth day to you." You'll notice that you divide Happy into two instances of the same note as you distinctly separate it into "Hap" and "py" during the song. Next, you sing "birth" but for the same duration as the time to do the two notes for "Hap" and "py." It's likewise for "day" and "to," but when you sing "you," that note is held for four times as long as "Hap" and "py," and twice as long as "birth," "day," and "to." So, the timing is 1, 1, 2, 2, 2, 4.

You may have noticed that as you sang each note, you inserted a brief quiet pause between each note. This pause was about equal to the time of the shortest note (1). You might ask "1" what? Well, that depends on the

pace you want your tune to have. If you set 1 to one second, then the tune would sound like it was being played in slow motion. If you set it to 1/4 second (250 ms), then you'd find the pace to sound more like what you would sing. Let's play this.

A program looking at the first six notes and their associated duration would then play the c note for 250 ms, pause for 250 ms, play the c note again for 250 ms, pause for 250 ms, play the d note for 500 ms, pause for 250 ms, play the f note for 500 ms, pause for 250 ms, play the e note for 1,000 ms and pause for 250 ms.

The Arduino *toneMelody* example (that you can find in the Arduino IDE [integrated development environment] under File/examples/0.2Digital/toneMelody) does something similar to what we just discussed in that it reads the note for the tune array, turns on the tone, and then delays for the duration. That's cool and we will do this in a lab, but there is one very serious caveat for playing tunes this way: You can't be doing anything else because the *delay()* function completely occupies the CPU while delaying for the indicated duration!

Fortunately, there is a better way. We can use an interrupt to play the tune in the background. For this, we will keep our tune in a single array, and have the interrupt

SUPERIOR EMBEDDED SOLUTIONS



DESIGN YOUR SOLUTION TODAY
CALL 480-837-5200

www.embeddedARM.com

TS-7670 and TS-7680 Industrial Computers

- Up to 454 MHz ARM w/ 256 MB RAM
- 2 GB Flash Storage
- Industrial Temperature (-40 °C to 85 °C)
- DIO, CAN, Modbus, RS-485

TS-7670 Features:

- GPS and Cell Modem
- 1x Ethernet
- 2x microSD Card Sockets

TS-7680 Features:

- WiFi and Bluetooth
- 24 VAC Power Input
- 2x Ethernet

Pricing starts at
\$129
Qty 100

\$168
Qty 1

Low cost plastic enclosure available



NEW! TS-4900 High Performance Computer Module

- Up to 1 GHz Quad Core ARM CPU
- Up to 2 GB DDR3 RAM
- WiFi and Bluetooth
- 4 GB Flash and microSD
- Gigabit Ethernet
- SATA II and PCI-Express
- DIO, CAN, COM, I2C, I2S
- Industrial Temperature (-40 °C to 85 °C)
- Supports Linux & QNX
- Android & Windows Coming Soon

Pricing starts at
\$99
Qty 100
\$134
Qty 1



TS-8150 PC/104 development board available

TS-TPC-8390 touch panel computer available



We've never discontinued a product in 30 years



Embedded systems that are built to endure



Support every step of the way with open source vision



Unique embedded solutions add value for our customers

check that array every 250 ms to play the tone in the background for 250 ms until the next interrupt occurs. A quarter of a second (250 ms) may not seem like much to us, but to a CPU processing instructions at roughly 16,000,000 per second, it can do 4,000,000 instructions while that tone is being generated in the background by one of the peripheral devices. The CPU can get a lot of other work done while humming a tune — if you use interrupts and let a peripheral do the humming.

But Wait, What's This 'Array' of Which You Speak?

Yikes, we are in the middle of learning about making music and suddenly we are using arrays — what the heck is an array? An array is a way of storing some related data in a location in memory so that it is easy to get to. An example of ‘some related data’ would be that list of notes for the Happy Birthday tune. There are 25 notes that we need to store somewhere in the computer memory, so we can pull them out one at a time when it comes time to play that note.

An array is a way of defining blocks of memory with each byte located in a continuous sequence of memory locations. So, for our song, we will set aside 25 memory locations, and then we can access each byte in that array by getting the byte from the first location, adding one to that location to get the next byte, and so on till we have added 25 to the first memory location and gotten the 25th note. Arrays are a data type, and the compiler knows how to handle them when you define them correctly.

The *toneMelody* example that comes with the Arduino IDE I mentioned previously plays the simple tune, *Shave and a Haircut, two bits*. In the code, you’ll see before the *setup()* function:

```
#include "pitches.h"

// notes in the melody:
int melody[] = {
  NOTE_C4, NOTE_G3, NOTE_G3, NOTE_A3, NOTE_G3, 0,
  NOTE_B3, NOTE_C4};

// note durations: 4 = quarter note, 8 = eighth
// note, etc.:
int noteDurations[] = { 4,8,8,4,4,4,4,4 };
```

First, we see that we include the *pitches.h* file which has definitions for musical note frequencies. For example, it defines:

```
#define NOTE_C4 262
```

This provides the actual frequency for the musical note, C4. A timer peripheral is used to generate a tone at the indicated frequency. We will discuss this in more detail later, but for the moment we want to understand the array data type that the notes are stored in:

```
// notes in the melody:
int melody[] = { NOTE_C4, NOTE_G3, NOTE_G3,
NOTE_A3, NOTE_G3, 0, NOTE_B3, NOTE_C4};
```

This statement defines an array named *melody* that stores integers (remember that *NOTE_C4* is an alias for the integer 262). In order to play the tune, we will want to extract each element in sequence one at a time to play each note. We do this as follows:

```
// define the variable to hold a note
int myNote = 0;

// set myNote equal to the number 3 element of
// the array
myNote = melody[3];
```

Arrays use computer style numbering so the first element of the array is stored at the 0 location, the second at 1, the third at 2, and so on. In the above example, *myNote* will equal the number 3 position in the array which is the third note, *NOTE_A3*. The first note in the array, *NOTE_C4*, is in the number 0 position.

Let’s review this. Arrays are blocks of memory with the first memory location in that block identified by the array name. (In our example the ‘*melody*’ term is an alias for the location of the first element in the array — *NOTE_C4*). Again, arrays are numbered beginning with 0, so when you request an element of an array you must remember this and request the element by subtracting 1 from the location of the element if you use normal arithmetic counting. (In our example, the third element in the array is indicated by *melody[3]*, *NOTE_A3*). There is no magic in this — it is just the way that arrays are done.

We might use that melody array as follows:

```
int i = 0;
int myNote = 0;

for(i = 0 ; i < 7 ; i++)
{
    myNote = melody[i];
    playTone(myNote);
}
```

In this case, we know the array has seven elements, so we use a *for-loop* to move through the array beginning at element 0 and continuing through element 6 (the first through seventh elements of the array). The actual *toneMelody* program uses a more efficient but less clear way to play the melody, but the underlying principle for using an array is the same. This may feel a bit iffy, but the labs will help reinforce the array concept.

Making Sounds With a Piezo Element

We are advised to make a joyful noise, and what better way than with a piezo element? Well, honestly, just about anything would be better — you don’t get much more low fidelity than this. Even calling the sound it

makes “noise” is being generous — so let’s ask, what’s cheaper? Now we’re getting somewhere, since these things are cheap and don’t require any external amplification circuitry.

Sound Components, Schematic, Layout

The piezo speaker included in the Arduino 101 projects kit (available through the NV Webstore) comes with long straight legs. You should bend its legs as shown in **Figure 3** so that it can be used conveniently on the Arduino proto shield breadboard (also shown in **Figure 3**).

We utilize the PWM peripheral that we discussed for using to control the angle of servomotors. You can set these PWMs so that they generate an audio frequency associated with a musical note. For instance, to generate the c musical note, we create an output waveform (see **Figure 4**) that turns on and off with a frequency of 261 cycles per second. Each of these on/off cycles occurs in $1/261$ of a second, or 0.003831 seconds.

Since we are dealing with microseconds, we multiply this by 1,000,000 to get 3,831 microseconds per cycle. Since we need to cycle the pin (turn the pin on and off) in that time, we turn it on for $3831/2 = 1,915$ microseconds (throwing away the fractional part) and off for 1,915 microseconds giving us a total of 3,830. We lost one due to our not wanting to use fractions, but who is going to miss a microsecond? Yeah that sounds like a lot of work for the computer, but fortunately the CPU can simply tell the PWM to do this, and it does it in the background with no further intervention by the CPU until time to stop outputting that sound.

Sound Using Interrupts

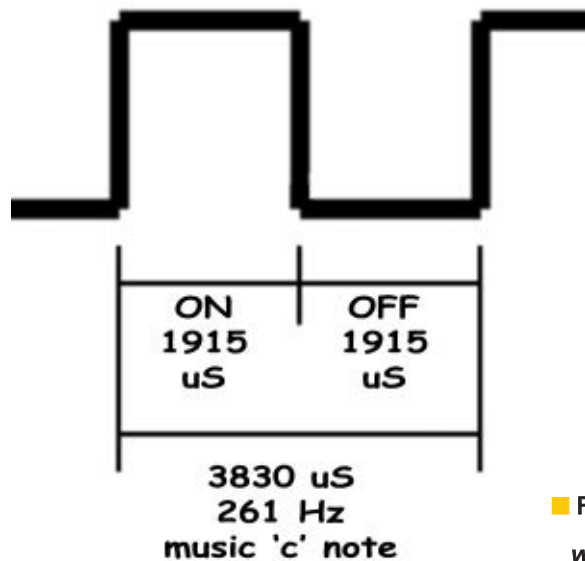
Earlier in this chapter, we discussed the Arduino example program, *toneMelody* that plays a sequence of notes and uses the *delay()* function to time the notes. The main problem with the methods shown in the *toneMelody* example from the Arduino IDE is that it uses the *delay()* function to time the duration of the tone in the melody.

The *delay()* blocks the CPU while playing a tone, so it can do nothing but wait for the tone to finish before it does the next thing. That doesn’t matter in the *toneMelody* example since all the Arduino is doing is playing a tune. If you want to do anything else while you play a tune, however, you’ll need to use interrupts. As discussed above, by using interrupts the CPU can do millions of operations while each tone of a tune plays in the background.

We want to have the CPU set the duration in a



■ **FIGURE 3:** Piezo with legs bent for breadboard.



■ **FIGURE 4:** A c note waveform.

peripheral timer and have that timer interrupt the CPU when the duration is passed, so that it can stop the current tone and then instruct the PWM to play the next tone. We will generate tones using the PWM and keep track of the duration of the tone using a timer interrupt. The duration of each tone will vary depending on the tune being played, but each duration will be in common units. For instance, we might have a melody with 100 mS units of duration. The shortest time we can play a tone will be 100 mS, and each longer interval will be some number of those periods.

We then keep an array of durations which tells us how many of these periods to play a tone. For instance, if in my tone array at position 10 we find a c note, we then look in the duration array at position 10 and see that it is 5. Since we know that the minimum duration is 100 mS, we will start that c note on the PWM and set the timer to interrupt the CPU when 5×100 mS have passed; that is 500 mS, or half a second. In order to play a melody this way, we will need the tone array, the duration array, the number of tones we are going to play, and the minimum duration of a tone.

For example, we use the following data to play Happy Birthday:

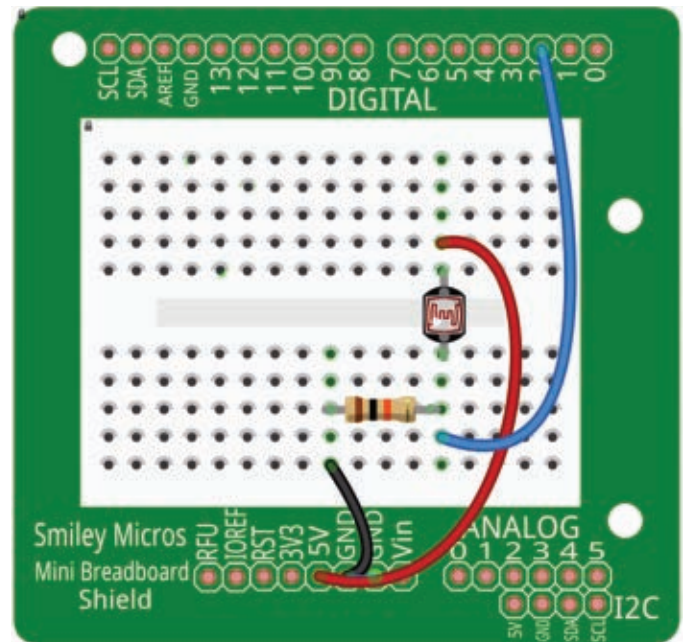

```

int timerMS = 40;
int dataCount = 50;
int freq[] = {587,0,587,0,659,0,587,0,784,0,740,
0,587,0,587,0,659,0,587,0,440,0,784,0,587,0,587,0,
1175,0,988,0,1568,0,1480,0,1319,0,1047,0,1047,0,
988,0,1568,0,880,0,1568,0,};
int duration[] = {3,1,2,1,8,1,8,1,8,1,16,
1,3,1,2,1,8,1,8,1,8,1,16,1,3,1,2,1,8,1,8,1,8,1,8,
1,16,1,3,1,2,1,8,1,8,1,8,1,16,1,};

```

We read each note and duration until we have read all the data as indicated by `dataCount` which shows we have 50 tones to generate. For each element in the `freq[]` and `duration[]` arrays, we set the PWM to play the tone and then the timer to interrupt after a certain number of base durations have elapsed. For instance, the first tone is 587 and the first duration is 3. So, we set the PWM to output a square wave with a frequency of 587 Hz and set the timer to $3 \times 40 \text{ ms} = 120 \text{ ms}$.

The 587 Hz tone will sound on the piezo element for 120 ms. The next tone is 0, which is a silent space between tones and it plays for one duration. The third element is also 587, but this time we only play it for $2 \times 40 \text{ ms}$. We do this process 50 times as indicated by `dataCount`, then the tune is finished.



■ FIGURE 5: *Light sensor interrupt breadboard.*

Lab 1: Sensor Hardware Interrupt – Light

We will use what we learned in Chapter 10 about light sensors and utilize a light sensor to generate a hardware interrupt. With this, we can interrupt the Arduino CPU by simply waving our hand over the board – no need to touch anything! (You may want to refer back to Chapter 10/Lab 1 if the circuit isn't clear.)

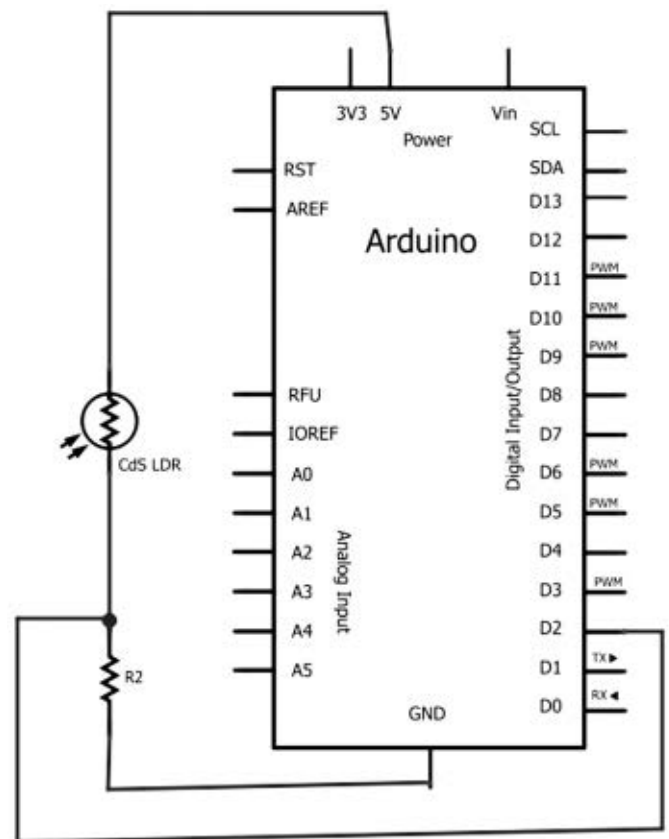
Parts required:

- 1 Arduino
- 1 USB cable
- 1 Arduino proto shield and jumper wires
- 1 CdS light sensor
- 1 10,000 Ω resistor

Estimated time for this lab: 30 minutes

Check off when complete:

- ☐ Build the circuit shown in **Figures 5** and **6**. This circuit is nearly identical to the one we built in Chapter 10, except that instead of having the sensor connected to the A0 pin for doing ADC, we connect it to pin 2 for doing an external hardware interrupt.
- ☐ Load the following program into the Arduino IDE. (All code files are available for download at the article link.)



■ FIGURE 6: *Light sensor interrupt schematic.*

```

// A101_ch9_light_sensor_voltage 9/28/14
//Joe Pardue

#define INT0 0 // INT0 is on pin 2 in the UNO

```

```

// You must use volatile when using variables in
// interrupts
volatile int state = 0;

```

```

unsigned long time = 0;
int count = 0;

void setup() {
  Serial.begin(57600);
  Serial.println("Measure light sensor voltage
  rev 1.0");
  attachInterrupt (INT0,lightInterrupt,RISING);
}

void loop() {
  // The time statements are used to prevent
  // multiple interrupts caused by a bouncing
  // signal on the pin
  if(state)
  {
    if((time+5) < millis())
    {
      Serial.print("Light interrupt #:");
      Serial.println(count++);
      time = millis();
    }
    state = 0;    // reset state to 0
  }

  // This function is called when an external
  // interrupt occurs on the pin indicated by
  // attachInterrupt
  void lightInterrupt()
  {
    state = 1; //!state;
  }
}

```

- ❑ Compile and run the program, then open the serial monitor.
- ❑ Make sure the board is lit by a bright overhead light, and then pass your hand back and forth over the sensor. You should see output similar to that shown in **Figure 7**.

Lab 2: Timer Interrupt

Parts required:

- 1 Arduino
- 1 USB cable

Estimated time for this lab: 15 minutes

Check off when complete:

- ❑ This lab requires no additional circuitry and can be run on a bare Arduino.
- ❑ Load this next program into the Arduino IDE.
- ❑ Take special note of the logic being used in this program. We have a variable set aside to act as a flag to tell the `loop()` function when an interrupt has happened. Once each second, the `myTimer1` interrupt function is called and sets the flag. The `loop()` checks the flag, and when it sees the flag is set, one second has passed and it checks to see if the tick flag is set. If it is, then it writes 'tick' to the serial monitor and sets the flag to 0, so the next time a second passes, it will see that tick is 0, print 'tock,' and set the flag to one.

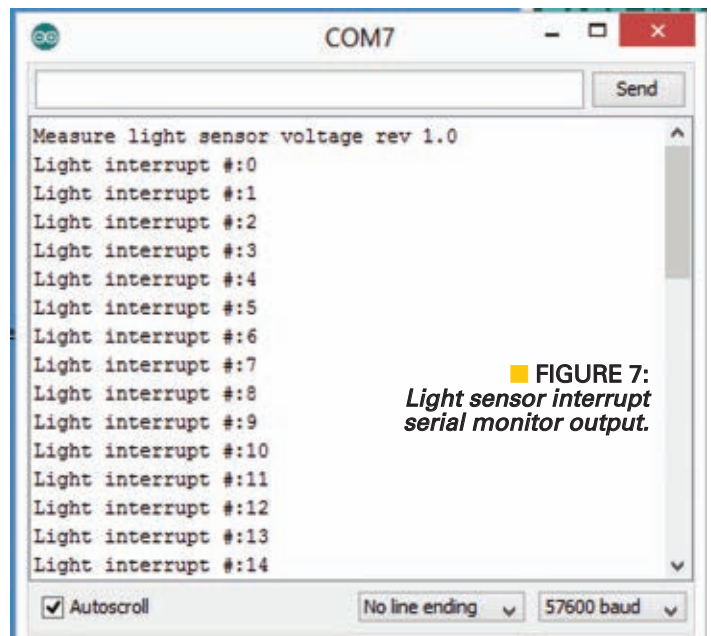


FIGURE 7:
Light sensor interrupt serial monitor output.

```

// A101_ch11_timer_interrupt 9/28/14 Joe Pardue

#include <TimerOne.h> // Timer 1 library

volatile int oneSecondFlag = 0;
int tick = 1;

void setup(){
  // Set up the serial port
  Serial.begin(57600);

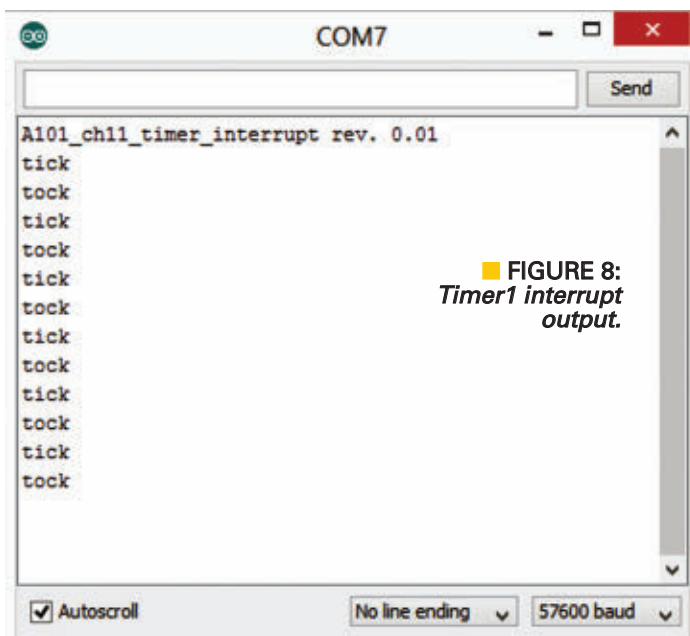
  // identify yourself
  Serial.println(F("A101_ch11_timer_interrupt
  rev. 0.01"));

  // initialize timer1 interrupt
  Timer1.initialize(1000000);
  // call it once per second
  Timer1.attachInterrupt(myTimer1);
}

void loop(){
  // if one second has passed
  if(oneSecondFlag)
  {
    oneSecondFlag = 0; // set the flag to 0
    if(tick)
    {
      Serial.println("tick");
      tick = 0;
    }
    else
    {
      Serial.println("tock");
      tick = 1;
    }
  }
}

// once per second this is called and sets the
// flag to 1
void myTimer1()
{
  oneSecondFlag = 1;
}

```

■ **FIGURE 8:**
Timer1 interrupt output.

- ❑ Compile and run this program. You should see the serial monitor output shown in **Figure 8**.

Lab 3: Finding the Piezo Resonant Frequency

We learned that these piezo elements are contained in a plastic container that will vibrate most readily at a specific resonant frequency, so that at that particular frequency the loudness of the sound is vastly magnified from sounds made at other frequencies. The resonant frequency is used when you want to create a very loud alarm sound.

Parts required:

- 1 Arduino
- 1 USB cable
- 1 Arduino proto shield and jumper wires
- 1 Piezo element

Estimated time for this lab: 30 minutes

Check off when complete:

- ❑ Build the circuit shown in **Figures 9** and **10**. We attach one leg of the piezo element to pin 6 of the Arduino, which we will use to generate PWM to create tones.
- ❑ Load the following program into the Arduino IDE:

```
/* A101_ch11_resonant_frequency 9/30/14
 * Joe Pardue */

#define tonePin 6
```

```
void setup() {
  Serial.begin(57600);
  Serial.println("A101_ch11_resonant_frequency
  rev 1.0");
}
```

```
void loop() {

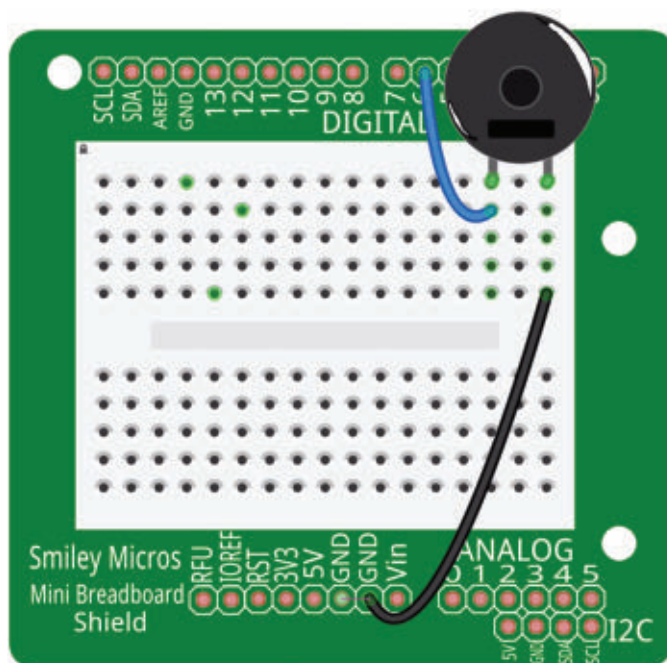
  for(int i = 0; i < 7000; i+=100)
  {

    tone(tonePin,i);
    delay(250);
    Serial.println("freq: ");
    Serial.println(i);
    tone(tonePin,0);
  }
}
```

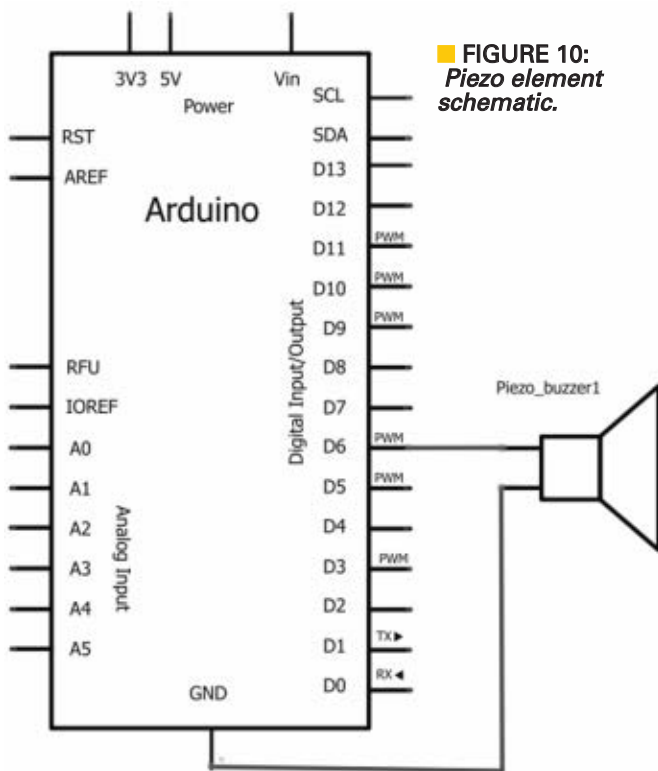
- ❑ Compile and run the program.
- ❑ Open the Arduino serial monitor as shown in **Figure 11** and observe the frequencies to judge the loudest sound.
- ❑ I observed that the loudest sound was around 4,500 — you may find a different frequency.
- ❑ Modify your code to bracket the observed frequency with 500 below and 500 above, and step through it in 50 Hz increments as shown next. Again, note that my use of 4,500 is not necessarily what you will observe.

```
void loop() {

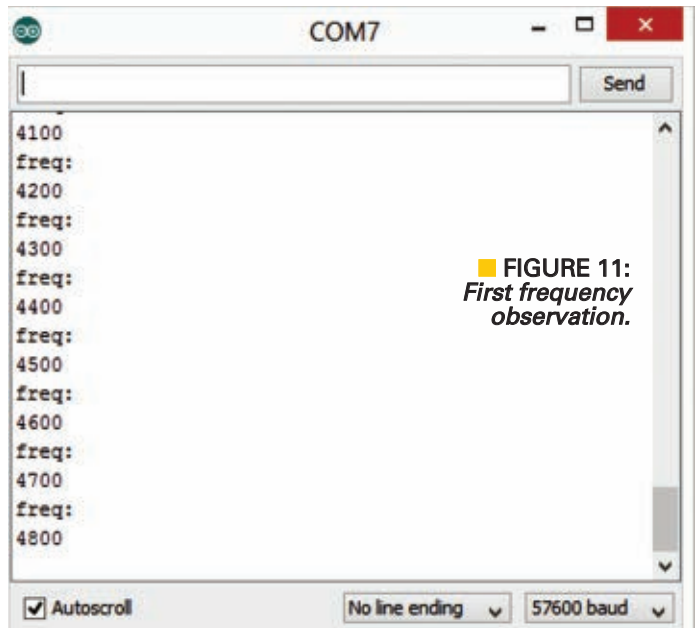
  for(int i = 4000; i < 5000; i+=50)
  {
    tone(tonePin,i);
    delay(250);
    Serial.println("freq: ");
    Serial.println(i);
    tone(tonePin,0);
  }
}
```



■ **FIGURE 9:** *Piezo element breadboard.*



■ FIGURE 10:
*Piezo element
schematic.*



■ FIGURE 11:
*First frequency
observation.*

In the next chapter, we will finish the Arduino 101 series by bringing together everything we have learned and building a stand-alone data logger. **NV**

- ❑ Compile and run the code to observe the loudest frequency.
- ❑ As stated, I observed the loudest frequency to be around 4,500.
- ❑ Next, bracket 200 Hz around your loudest frequency in 10 Hz increments. I used the following code:

```
void loop() {
  for(int i = 4400; i < 46000; i+=10)
  {
    tone(tonePin,i);
    delay(250);
    Serial.println("freq: ");
    Serial.println(i);
    tone(tonePin,0);
  }
}
```

- ❑ To me, 4,530 Hz seemed loudest. Differences in piezo elements and your hearing acuity may cause you to find a different frequency.

Well, we ran out of space this month. However, as a surprise bonus, you will find three additional labs at the article link that will let you generate alarms and music with your Arduino and a piezo speaker. In fact, the code used in Lab 6 allows you to play renditions of 14 different tunes.

Electronics Courses

Cleveland Institute of Electronics

Train at home to be a professional electronics or computer technician! Learn with hands-on labs, instructor support, online exams, videos and instructor led chat rooms.

FREE course catalog www.cie-wc.edu or (800) 243-6446

- **NEW!** Robotics Automation Lab
- **NEW!** Computer Security Specialist
- Industrial Electronics with PLC
- Broadcast Engineering
- Electronics with FCC
- PC Troubleshooting
- Electronics Troubleshooting
- Networking

Visit www.cie-wc.edu and start today!

CIE: 1776 E. 17th St., Cleveland, OH 44114 • (800) 243-6446 • Registration 70-11-0002H

ELECTRONET

CORIDIUM
Floating point **BASIC** for
ARM controllers from \$5.00
www.coridium.us

GET THE NUTS&VOLTS DISCOUNT!
Mention or enter coupon code **NVRMZ142**
and receive 10% off your order!
ramsey
www.ramseykits.com
AM/FM Broadcasters • Hobby Kits
Learning Kits • Test Equipment
...AND LOTS OF NEAT STUFF!

USB Add USB to your next project--
It's easier than you might think! **DLP Design**
USB-FIFO • USB-UART • USB/Microcontroller Boards
RFID Readers • Design/Manufacturing Services Available
Absolutely NO driver software development required!
www.dlpdesign.com

HOBBY ENGINEERING
Kits, Parts and Supplies
www.HobbyEngineering.com

MYRO www.myropcb.com
PCB, PCBA and More
LOW cost with HIGH Quality
1-888-PCB-MYRO

SERVO MOUNT GEARS
METAL & PLASTIC @ ServoCity.com
Patented
STARTING AT ONLY \$3.08

THE ORIGINAL SINCE 1994
PCB-POOL
Beta LAYOUT
• Low Cost PCB prototypes
• Free laser SMT stencil
with all Proto orders
WWW.PCB-POOL.COM

superbrightleds.com
COMPONENT LEDs • LED BULBS • LED ACCENT LIGHTS

\$3
2 Layers ea.
• FULL DRC CHECK
• CAD/CAM REVIEW
• SOLDER MASK ON BOTH SIDES
Accutrace inc.
www.PCB4u.com
(408) 748-9600
PRINTED CIRCUIT BOARDS

INVEST in your BOT!
HITEC
12115 Paine Street • Poway, CA 92064 • 858-748-6948 • www.hitecusa.com

Got electronics?
www.nutsvolts.com

ALL ELECTRONICS CORPORATION
Electronic Parts and Supplies.
www.allelectronics.com
Free 96 page catalog 1-800-826-5432

For the ElectroNet
online, go to
www.nutsvolts.com
click **Electro-Net**

MOBILE APP NOW AVAILABLE!
SERVO
Download NOW On Your
Favorite Mobile Device!
FOR ROBOT BUILDERS
iOS • ANDROID • KINDLE FIRE

2-For-1 Holiday Offer



Give a one year (12 issues) subscription to a friend and receive a *FREE* one year subscription for another friend, or yourself!

Not only will they get a full year of Nuts & Volts delivered right to their door, they'll also get free access to our digital (web) edition, PLUS our mobile (iOS, Android, Kindle Fire) edition.

Subscribe online at:
www.nutsvolts.com

or call:

877-525-2539 (toll free)
818-487-4545 (outside US)

Be sure to use promotion code
Y4WXMS when ordering

U.S. Price: 1 Yr - \$26.95

Bridging the Gap of Packaging With 3D Printing

From 2006 through 2009, I wrote a column for this magazine called *Getting Started with PICs*. It was intended to introduce the programming of microcontrollers to many who were still using 555 timers and CMOS/TTL chips in their designs, and help those just getting started with micros. The main point was it's not that hard to get started, and I wanted to help. I was offering some guidance through the various projects I introduced or topics I covered. Well, I'm back to do the same thing ... only this time for 3D printing.

As any *Nuts & Volts* reader probably knows, there are three main areas to designing an electronic project: hardware, software, and packaging. The hardware can include not only the components but also the circuit board, schematic, and breadboarding. There are now many different software options to create schematics, circuit boards, or even virtual breadboard designs. Many of these options are free or very low cost. So, hardware has really come a long way.

Creating microcontroller software has also come a long way. There are numerous free compilers available for the C language, BASIC language, and many others. Plus, you can get hardware programmers for loading the software into the microcontroller for easily under \$25, or modules with USB bootloaders built in for less than that. Code samples are essentially everywhere via the Web.

So, the last remaining difficulty seems to be the

packaging. I'm talking about the case or the box that the electronics are packaged into. This has been a difficult and sometimes expensive part of a completed design. I've seen some very talented people make some excellent cases for their creations, but the hours put into it and the customization makes it difficult to reproduce. This is similar to the early days of home circuit board design when every one was hand drawn and etched in a copper eating solution.

I've spent many years searching for a better way to make the packaging for my designs. I wanted an easy way to both create designs and reproduce them. I tried using various Computer Aided Design (CAD) software packages which I still find difficult to use unless you work with it everyday. To produce my designs, I needed access to CNC machines or laser cutters. I found those through various companies who offered this service online, and now many local makerspaces can fill this need (for a sometimes costly fee).

I also found that working in 2D with CNC or laser was limiting my box creations. I started with a simple clear plastic top and bottom long before there was the "sick of beige" design by dangerousprototypes.com, but I wanted more than that. During all this, the home 3D printer revolution was launched, but initially it looked very complicated. Everything was a complex kit.

The software to control it was buggy, and there seemed to be a new version popping up everyday. It seemed way too confusing. So, I sat back and watched it grow, hoping a more complete package of software and hardware would evolve.

After about two years of just watching the 3D printer designs improve while at the same time watching the cost come down, I was ready to buy my own 3D printer. I had a budget of just under \$1,000, and I wanted a machine that was fully assembled with included software designed to work with it. I soon realized I also needed a CAD type software package to design the 3D product to build on the 3D printer; nobody seemed to have put all that together — at least not at a price I could afford.

Then, I discovered a free software call Tinkercad. It seemed easy to use since you just place a variety of shapes together to form your object, then group them

HELPFUL INSTRUCTIONS TO CREATE AND PRINT

Post comments on this article and find any associated files and/or downloads at www.nutsvolts.com/index.php?/magazine/article/December2014_GettingStarted3DPrinting.

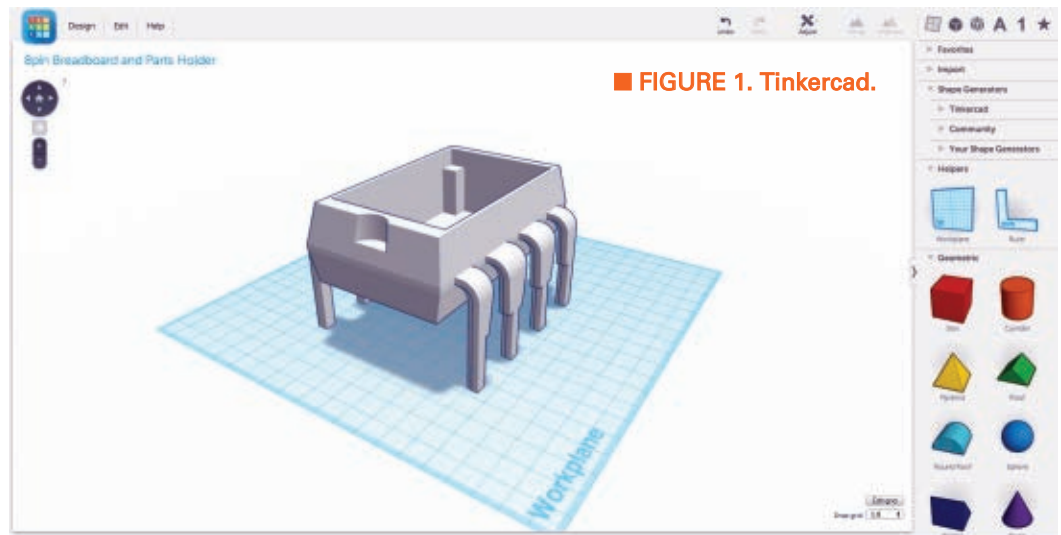
together as one solid piece. I worked through all the Tinkercad online tutorials — which are great at teaching you how to use the software in a step by step method — to create various 3D designs. It didn't take long to get comfortable with this software, so I started designing my own creations.

From there, you can export your design as a .stl file. Any 3D printer can print the design, including professional 3D printer companies such as Ponoko or Shapeways. Then, unfortunately, news came out that Tinkercad was going to shutdown. I was very disappointed, but soon Autodesk — the makers of many popular CAD software programs — stepped in to take it over, keep it alive, and added improvements. I had found my easy-to-use design software!

You can only create designs for so long before you actually want to build something. It was time to purchase a 3D printer.

I shopped around and found I had several choices for under \$1,000. I was leaning towards a low cost Printbot Simple or Solidoodle printer. I was ready to purchase a Printbot Simple as a starting point even though it didn't print ABS or have a heated bed. However, the price was good and I wanted something easy to use. They offered an assembled version and appeared to have some kind of software setup and manual that worked well together based on feedback.

When I went to the website to buy it, I noticed the price had gone up \$50 in the few days since I last looked at it. That was like a 17% increase which was still in my budget but now made other options worth reconsidering, so I held off. Solidoodle was also a possibility for \$499, but there were extra cost options that I wasn't sure I needed or not, so again I delayed. While searching further, I discovered the Davinci 3D 1.0 printer. This would be my huckleberry.

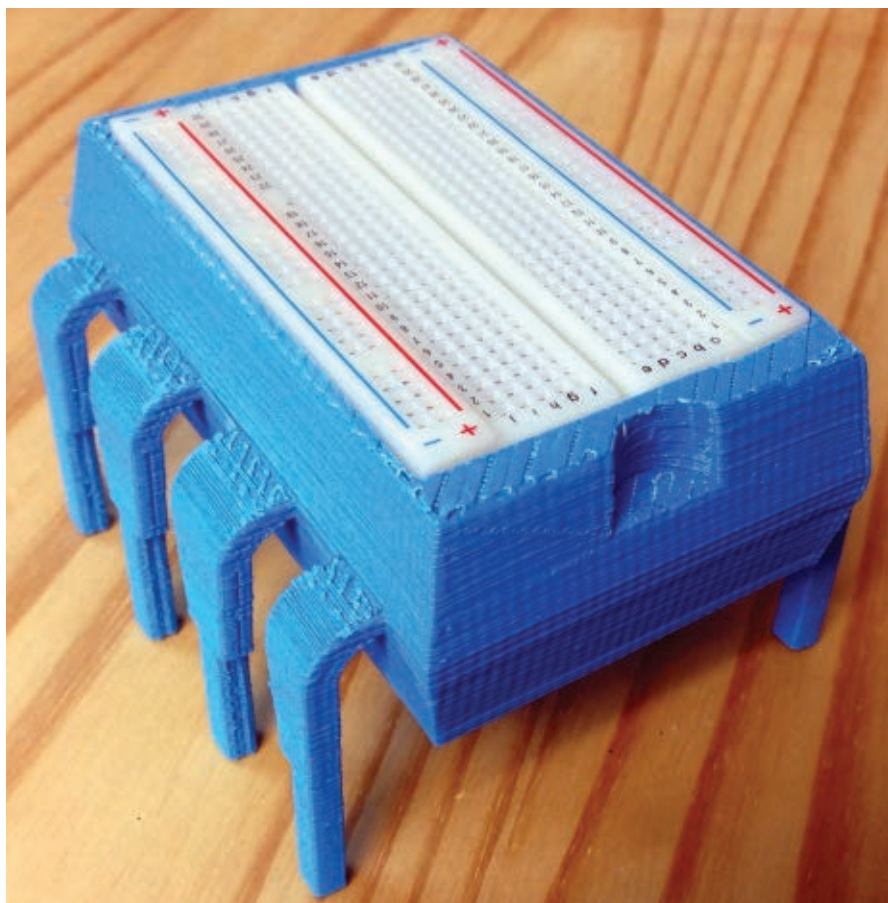


■ FIGURE 1. Tinkercad.

The Davinci 3D printer was released in early 2014. It's fully assembled and in a completely enclosed plastic case. It came with its own software, had a large 7.8" x 7.8" x 7.8" heated build platform, and — best of all — was just



■ FIGURE 2. Davinci 3D printer.



Resources

My website and blog:
www.elproducts.com

My YouTube Channel:
www.youtube.com/user/beginner_electronics

My 3D designs:
www.thingiverse.com/elproducts/designs

Tinkercad:
www.tinkercad.com

Davinci 3D:
us.xyzPrinting.com

Davinci Forums:
www.solidforum.com
www.voltivo.com

■ **FIGURE 3. Large eight-pin IC breadboard holder and component box.**

under \$500. The initial reviews were negative because it was not open source and required you to buy their filament cartridges. Of course, it was quickly hacked and users were running third party software on it.

I didn't want to hack it. I wanted to use it as-is. It didn't take long before reviews came in from users who said it printed just fine with the included software. That's what I wanted to hear!

It was clearly popular as it sold out often on Amazon, but after about a month I finally was able to get one. I received my first 3D printer — the Davinci 1.0 — and within an hour, I had completed my first 3D print. It was a sample print of a keychain token which the Davinci has as part of its three built-in sample print selections. It went so smooth, I was hooked.

From there, I started to print some of my Tinkercad designs. The designs came out excellent and I was soon designing new items to print. I felt like a kid with a new toy. I also found other users at a couple forums. Many were already hacking their Davincis, but a few people were having fun like I was — without modification.

After several weeks of printing, I discovered that even though the Davinci was designed to make 3D printing "out-of-the-box" easy, I could improve it with just a few

adjustments. I found that the auto-calibrating base still left the prints lifting off the heated bed at times. I decided to manually adjust it myself to my own settings, and soon found the prints were coming out much better and sticking to the heated platform during the print without warpage or lifting.

Like any tool, you have to learn some of the "features" to get it to work the way you want. It was really just several minor tweaks to fit my needs. I then began to design my first 3D case for one of my electronic designs. I created a case for my homemade PICKit 2 clone programmer design I feature on my website at www.elproducts.com. Frankly, it's probably one of the ugliest case designs I've made, but I learned a lot from it. After about 15 prints, I was able to get everything to fit together properly and even designed in a snap feature for the top. My skills using Tinkercad continued to get better as I could now see the direct results of my designs. I'm now to the point where creating a successful design can be completed in one or two prints.

With the 3D printer firmly planted in my workshop, I began to search Thingiverse.com for designs I could build. I found a plastic version of an eight-pin integrated circuit in actual size, but I decided to make it bigger. In fact, I made

it big enough to hold a small breadboard and have storage space inside to hold all the components and programmer (of course, for someone to use with my book, *Programming PICs in BASIC*).

Several people I showed it to thought it was very interesting, so I decided to share the fun I was having through YouTube videos. I launched a channel at youtube.com/user/beginnerelectronics to showcase how to get started with both electronics and 3D printing.

It was clear to me that the final step of hardware, software, and packaging was solved, and many people were sharing designs on Thingiverse and elsewhere. Designing custom packaging for electronic creations and then easily sharing the designs with anybody who wanted to print them on their own 3D printer was here.

I know there are still many who are still trying to decide if this 3D printing is worth the expense, and some are still confused about where to start. That's why I teamed up with *Nuts & Volts* again to launch this column.

The goal of this column is all about helping you get started with 3D printing for your electronic designs. Over time, we'll include projects that will be first released to *Nuts & Volts* readers — mostly related to packaging of electronics. However, I will also show 3D printed tools or objects for your electronics lab, and helpful tips and tricks. I will try to cover anything new that may make 3D printing easier in the future, as well. The idea is to help you get started with 3D printing so you can create your own 3D designs.

To make things less expensive and easier for the beginner, I'll be using my low cost Davinci 1.0 printer and the free Tinkercad software to produce the .stl files to print. This allows anybody to get started and follow along with me for less investment. Any of the designs I produce for this column will be shared through my website

and also Thingiverse after the article is published. In most cases, the designs will work on other printers as well, so you can import the .stl file into any software you want to print or modify it in as you see fit.

If you have ideas for 3D prints you'd like to see here, send me an email at chuck@elproducts.com. It's great to be back at *Nuts & Volts* to share in the fun of building electronics and 3D printing with the readers. **NV**

ALL ELECTRONICS

www.allelectronics.com

Order Toll Free 1-800-826-5432

ULTRACAPACITOR 1500 FARADS - 2.7 VDC

Maxwell Technologies # BCAP1500 P270. Ultra-high capacitance for extended power availability in a variety of situations. Allows function to remain available during power dips and outages, relieves batteries of burst power functions. High current capability, 100 Amps. Threaded stud terminals, M12-1.75. 85mm L x 60mm D (3.34" x 2.36" D), excluding terminals. **\$25⁰⁰ each**
CAT# UCP-1500



PUSH-ON MOMENTARY SWITCH

C&K # 8531. Single-pole, momentary-on pushbutton switch. Rated 1A 120Vac. 15/32-32 threaded bushing. 0.33" diameter red pushbutton. Sealed epoxy body. Solder turret terminals. USA made. UL. **\$2⁵⁰ each** **10 for \$2.25 ea.**
CAT# MPB-39



2-8 VDC MINI-MOTOR

Mabuchi # FF-030PA 08250. Body: 15.5 x 12 x 20mm long. Shaft: 1.5mm diameter x 6mm long. Two tapped mounting holes on 10mm centers, ISO M2.0 x 0.4. No-load rating: 6Vdc, 41mA, 10,500 RPM. Solder terminals. **\$1²⁵ each**
CAT# DCM-433
100 for \$1.05 each



5K LINEAR POTENTIOMETER

Bourns #PCL1J-C28-CAB502. Single-turn, 0.75W potentiometer. 6mm diameter flatted plastic shaft. 9mm diameter threaded plastic bushing. 22mm square x 7mm. **\$1⁵⁰ each**
CAT# PPT-5KL



FLOAT SWITCH WITH INTEGRATED THERMISTOR

Horizontal/side mount liquid level float switch. For sensing rising or falling liquid levels. Hermetically sealed magnet and reed switch for accurate, reliable long-life operation. Integrated 10K ohm NTC thermistor allows monitoring of liquid temperature as well. Threaded 13/16" male Nylon stem. Polypropylene float, 2.4" long from hinge to end. **\$6⁷⁵ each** **10 for \$6.25 each** **100 for \$5.90 each**
CAT# FLW-1



AUTOMATIC WIRE STRIPPER

A quick, easy way to strip a wide variety of insulated wires and coaxial cables. Squeeze the handles together to clamp on an insulated wire. Continue to fully squeeze the handle to activate the metal strippers to pull off the insulation. **\$5³⁵ each**
CAT# WSI-1

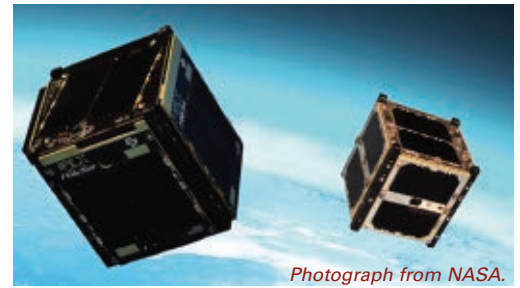


LASER POINTER & 12 LED POCKET WORK LIGHT

Nebo 6053 Larry 2. Combination laser pointer and extremely bright LED worklight. Magnetic 180° swivel clip for attaching the light to any magnetic surface and freeing up your hands. Water and impact resistant. Max laser output <5mW / 650 nm. 6.75" long. Operates on three AAA batteries (included). **\$16⁰⁰ each**
CAT# FL-40



CubeSats — Part 1



Photograph from NASA.

"Reach low orbit and you're halfway to anywhere in the solar system."

Robert Heinlein

Making Satellites

Space exploration is expensive. So much so, that for most of the "Space Age," governments were the only entities capable of footing the bill for "Where No Man Has Gone Before." However, universities and at least one high school (Thomas Jefferson High School for Science and Technology) have or are now designing, building, and launching their own satellites.

This month, I want to describe this new breed of satellites. Later, I'll

describe where to purchase kits and a model that I'm designing as a stepping stone to the real thing. I plan to call this model: My First Satellite.

Satellites in Education

Back in 1999, Professors Bob Twiggs and Jordi Puig-Suari utilized a class project to help their Stanford students become better aerospace engineers. They assigned them the task of building two Earth-orbiting satellites: Opal and Sapphire. The satellites were microsattellites — hexagonal airframes 12 inches tall, 18 inches across, and weighing 50 pounds. Their telemetry radios were off-the-shelf amateur radios with an output between three and five watts.

Over time, their students kept adding more subsystems to the two satellites. Gradually, the satellites became too expensive and far from ever being launched. Now, their

students would never complete this practical satellite application, much less get it off the ground. However, Dr. Twiggs and Puig-Suari recognized a solution hidden in the satellite named Opal. Opal contained picosatellites — tiny satellites that Opal would release after reaching Earth orbit.

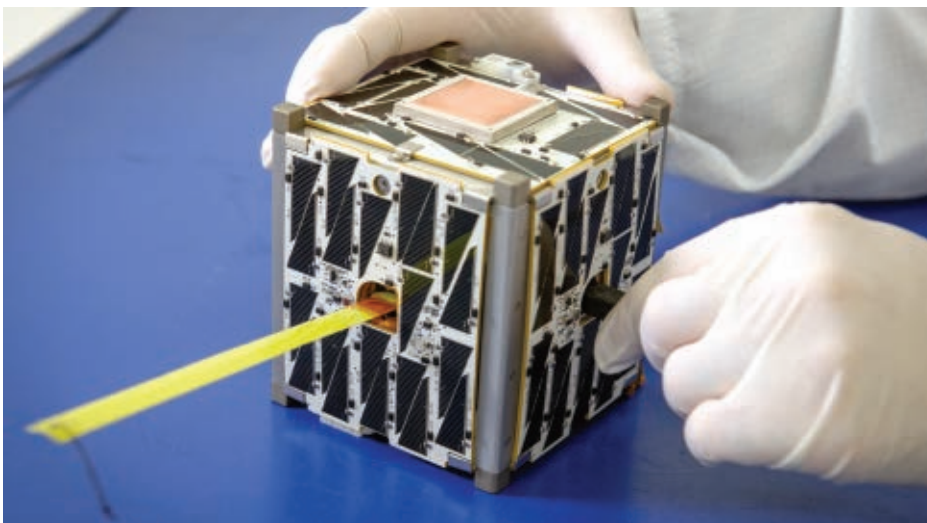
They used the limited volumes and weights of the Opal picosatellites as a standard to force their students to innovate small lightweight satellites. Because of the satellite's limited volume and mass, students were unable to expand the capability of the satellite. This left them with a satellite they could finish within two years, or during the length of their Master's program.

We call this standard the CubeSat and it's becoming more popular. However, it took some time. Initially, many in industry felt that CubeSats were toys at best. Today, most realize that CubeSats are an entirely new paradigm in satellite design with lots of potential. That potential even includes the exploration of distant planets.

The CubeSat Form Factor

The basic CubeSat is a 10 cm cube (for a volume of one liter) weighing no more than one kilogram. This is called a one unit, or 1U CubeSat. There are 2U and 3U CubeSats for satellites needing more avionics. These larger units are essentially stacks of 1U CubeSats.

The four vertical edges of a CubeSat are low friction metal rails to which the airframe sides are attached. The airframe is constructed from aluminum panels — a material



You really get a scale of a CubeSat's size in this picture. You can see solar cells protected in cover slips that are clipped to the airframe, along with one of the antennas in its deployed position. Image courtesy of NASA.

Post comments on this article and find any associated files and/or downloads at www.nutsvolts.com/index.php?/magazine/article/december2014_NearSpace.

that's easy to machine and that doesn't outgas in the vacuum of space. Mounted to the outside of the airframe are the CubeSat's solar cells and antennas.

CubeSats use metal tapes for their antenna elements. The antenna elements are lengths of metal measuring tape. The metal tape is flexible, so it can be bent back tightly against the sides of the airframe.

However, because of the curl along their length, the metal tape snaps back straight if it's not restrained. As a result, after releasing their restraint, the metal tapes snap straight out and wobble back and forth until their motion dampens out.

The antennas are tied to the bottom of the CubeSat with plastic fishing line prior to being loaded into their launcher. After the CubeSat's deployment, a timer triggers a circuit to melt the restraining fish line with a hot nichrome wire. This frees the ends of the measuring tape so they spring straight out, creating the CubeSat's antenna. I guess you could say that good things come in small boxes.

CubeSat Avionics

The avionics inside a CubeSat consist of a stack of PC/104 cards. According to the PC/104 Embedded Consortium, these cards are a smaller version of the ISA PC and PC/AT bus. They are too small for PC use; their small size makes them more appropriate for embedded systems.

Each PC/104 printed circuit card measures 3.55 inches by 3.775 inches on a side. There is no vertical insertion of cards like those that we find in a PC back plane. Instead, the cards are stacked together like Arduino shields using stackthrough connectors. These connectors have a female receptacle on top, and long male header pins on the bottom.

The stackthroughs are located on one side of each PCB (printed circuit board). There are also four mounting

holes: one for each corner of the PCB. The corners of the PC/104 cards are attached together using spacers. This makes the stack of PCBs more rigid and less likely to break apart.

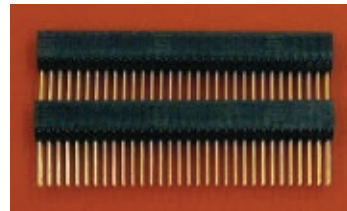
By the way, the number 104 in PC/104 comes from the fact that there are 104 pins in the stackthrough. The 104 pins are arranged into four rows. The first two rows contain 32 pins each and are for the eight-bit bus. The remaining two rows contain 20 pins each and are for the 16-bit bus.

Like PC cards, there are four bus voltages in the PC/104 standard: 12V, -5V, +5V, and +12V. Needless to say, ground (0V) is also included in the rows of pins via multiple redundant ground pins.

Launching CubeSats

The first CubeSat launch took place in June 2003, but not as the primary payload of an American rocket like the Atlas or Delta. That's because CubeSats are far too small to make enough money for rockets like these.

The first CubeSats were the payload of a repurposed Soviet era ballistic missile called the SS-19. Ten years later, we find CubeSats are launched by the larger rockets, but are used as rocket ballast. For proper performance, rockets need dead

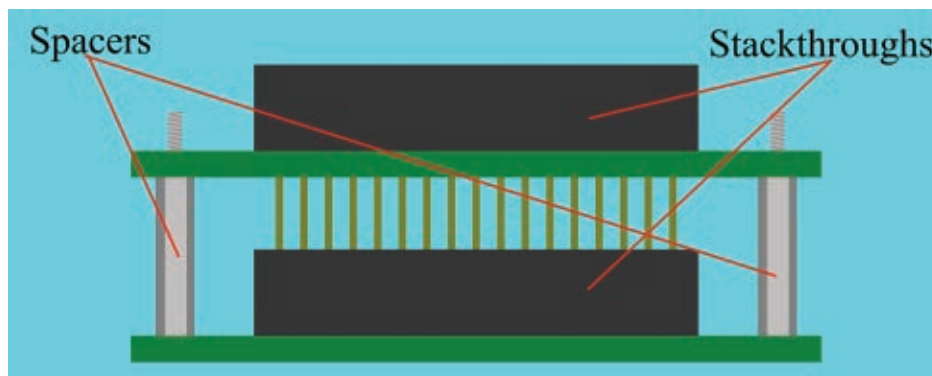


Two stackthrough connectors of the type I use in my projects. The long length of their pins ensures there's plenty of clearance between the PCBs for the components soldered to the boards.

weight (or ballast) to even out the distribution of their weight. Secondary payloads like CubeSats make more sense as ballast than a passive chunk of metal.

Remember Opal, the late 1990's microsatellite? Its picosatellite launcher became the basis for P-POD, or Poly-Picosatellite Orbital Deployer. The P-POD mechanism is a hollow square-shaped tube containing a spring operated ejection plate in its base. Each P-POD contains a stack of three CubeSats. All three CubeSats rest on top of the compressed ejection plate. At the time of release, a latch on the lid of the P-POD springs back, opening the top of the P-POD and letting the ejection plate push all three CubeSats out into space.

There's a switch at the base of each CubeSat. As long as this switch



A diagram showing how PCBs like those found in PC/104 cards are stacked together. The stackthrough connectors maintain the electrical connection between the PCBs, while the spacers maintain the structural integrity of the stack.



The 3U PW-Sat CubeSat being loaded inside its P-POD launcher. The door on the left side will be closed and latched shut after the CubeSat is safely inside. This student-built satellite is from the Warsaw University of Technology.
(This image was downloaded from their website.)

is held down — like when the CubeSats are stacked on top of each other — the CubeSats cannot begin operations, like antenna deployment and radio transmissions.

What's so awesome about the P-POD is that primary payload customers are normally reluctant to have other satellites ride along with the satellite. This makes sense when you are paying tens and perhaps hundreds of millions of dollars to launch your satellite into Earth orbit.

First, you don't want other

interference. That interference can result in monetary jamming or even inappropriate operation when the primary satellite receives the wrong command.

So, by keeping the CubeSats confined inside the P-POD and powered off until well after the main payload is released, customers of high priced satellites are more accepting of low value CubeSats riding the same rocket into space.

The low orbit at which most

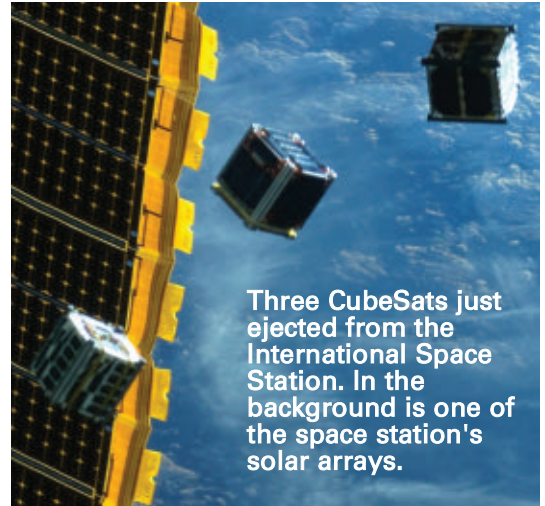
Satellites Come in Different Sizes

According to Rick Fleeter of Aero Astro, one way to classify satellites is based on their mass, or weight. Here's Mr.

Fleeter's category of satellites. As you can see, CubeSats fit within the picosatellite category.

Micro - 100 kg
Nano - 10 kg
Pico - 1 kg

objects colliding with your satellite. This is a real risk as there are instances of rocket stages bumping into satellites during their deployment. Second, satellites transmitting close to each other can create radio



Three CubeSats just ejected from the International Space Station. In the background is one of the space station's solar arrays.

CubeSats are released means they'll remain in orbit for a few months to a few years. This insures CubeSats won't become part of the space junk that endangers more expensive operational satellites.

Today, CubeSats are even being launched from the International Space Station (ISS). At the station's low orbit, a CubeSat launched from the ISS will only remain in orbit for about one year.

I found several good CubeSat articles while writing this piece. Online there's a nice interview with Dr. Twiggs at www.spaceflightnow.com/news/n1403/08cubesats/, and in the September 2014 issue of *Air and Space* magazine, you'll find a discussion about the shrinking space program created by the CubeSat revolution.

You can find additional information in the CubeSat forum at <http://cubesats.wikidot.com>. Finally, I want to thank Dr. Twiggs for answering my questions about CubeSats and their history.

Onwards and Upwards,
Your near space guide **NV**

Feedback Motion Control

The Old Way

- 1) Build robot
- 2) Guess PID coefficients
- 3) Test
 - 3a) Express disappointment
 - 3b) Search Internet, modify PID values
 - 3c) Read book, modify PID coefficients again
 - 3d) Decide performance is good enough
 - 3e) Realize it isn't
 - 3f) See if anyone just sells a giant servo
 - 3g) Express disappointment
 - 3h) Re-guess PID coefficients
 - 3i) Switch processor
 - 3j) Dust off old Differential Equations book
 - 3k) Remember why the book was so dusty
 - 3l) Calculate new, wildly different PID coefficients
 - 3m) Invent new, wildly different swear words
 - 3n) Research fuzzy logic
 - 3o) Now it is certainly not working in uncertain ways
 - 3p) Pull hair
 - 3q) Switch controller
 - 3r) Re-guess PID coefficients
 - 3s) Switch programming language
 - 3t) Start a new project that doesn't need feedback control
 - 3u) See parts in box. Feel guilty. Go back to old project
 - 3v) Start testing every possible combination of PID coefficients
 - 3w) Apply eye drops to red, blurry, sleep-deprived eyes
 - 3x) Wait. It's working!
 - 3y) Decide not to do any more projects that require control systems
 - 3z) Wonder why someone doesn't just make a thing that tunes itself

The Kangaroo x2 Way

- 1) Build robot
- 2) Press Autotune
- 3) Get a snack

Kangaroo x2 adds self-tuning feedback to SyRen and Sabertooth motor drivers.



\$24.99



www.dimensionengineering.com/kangaroo

CLASSIFIEDS

SURPLUS

SURPLUS ELECTRONIC PARTS & ACCESSORIES



Over 20,000
Items in Stock.

Cables
Connectors
Displays
Fans

Hardware
LEDs
Motors
Potentiometers

Relays
Semiconductors
Service Manuals
Speakers

Switches
Test Equipment
Tools
VCR Parts

Surplus Material Components
SMC ELECTRONICS
www.smcelectronics.com

No Minimum Order.
Credit Cards and PAYPAL Accepted.
Flat \$4.95 per order USA Shipping.

LIGHTING



KITS/PLANS



QKITS LTD
sales@qkits.com

1 888 GO 4 KITS

Arduino • Raspberry Pi
Power Supplies
MG Chemicals
RFID

\$8.50
flat rate
shipping



Visit us at:
www.qkits.com

Subscribe today!
www.nutsvolts.com

ROBOTICS

MaxBotix
High Performance Ultrasonic Rangefinders

Check us out Today!

4-20HR-MaxSonar[®]-WR™

- 4-20mA output
- High noise tolerance
- IP67 rated
- 1.5 mm resolution
- Multi-Sensor operation
- Calibrated beam pattern
- 8Hz read rate
- Ideal for industrial use



UCXL-MaxSonar[®]-WR™

- Great for design engineers
- Multiple mounting options
- Light weight industrial sensor
- Incredible noise immunity
- Smallest IP67 sensor in size
- 1cm resolution
- Automatic calibration
- Ideal for outdoor UAV's



Phone: 218-454-0766

Email: sales@maxbotix.com

www.maxbotix.com

SERVICES



Sandpiper Electronics Services LLC

"design of custom electronics instrumentation"

Circuit design & drawings, PCB layout, fabrication & population
Prototype development for research, industry and inventors.
39 years experience in electronics development for research &
flight applications, laser systems & laboratory research

FREE no obligation consultations

www.sandtronics.com

AUDIO/VIDEO PARTS

ICEpower MODULES

50W x 2 (50ASX2) or
170W x 1 (50ASX2BTL)
Just add AC & speaker(s)
\$99 USA \$119 Worldwide
www.classicaudioparts.com

PARTS
EXPRESS
YOUR ELECTRONIC CONNECTION™

OVER 18,000
ELECTRONIC PARTS
IN STOCK



Speakers



Components



Project Accessories



Tools and Tech Aids

Call or visit us online today to receive
your **FREE** copy of our 2014 catalog!

parts-express.com/nuts

1-800-338-0531

CONTROLLERS

Join The
INTERNET of THINGS
REVOLUTION



from \$119

TRI SUPER PLCs

Powerful & Easy Ladder
+ BASIC Programming
Ethernet integrated
MODBUS TCP/IP
DI/Os & AI/Os integrated

tel : 1 877 TRI-PLCS
web : www.triplc.com/nv.htm

TRI TRIANGLE
RESEARCH
INTERNATIONAL

Nuts & Volts 2014 CD-ROM

Pre-order now to reserve your copy!

This CD-ROM includes all of Vol 35, issues 1 - 12
in pdf format and is PC and Mac compatible.

Also included on the disc are the
2014 Media Downloads

Only \$19.95

Regular Price \$24.95

FREE 1st Class Mail Delivery

Estimated shipping in mid-December

Order from the Nuts & Volts Webstore @ store.nutsvolts.com

(Discount price ends December 31st)

GREAT FOR DIYers!

Electronics from the Ground Up: Learn by Hacking, Designing, and Inventing

by
Ronald Quan

Are you fascinated by the power of even the smallest electronic device? Electronics from the Ground Up guides you through step-by-step experiments that reveal how electronic circuits function so you can advance your skills and design custom circuits. You'll work with a range of circuits and signals related to optical emitters and receivers, audio, oscillators, and video.

Paper back 544 pages.

\$30.00



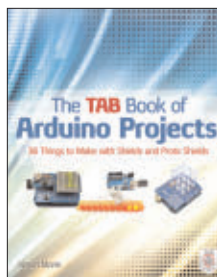
The TAB Book of Arduino Projects

by Simon Monk

The ultimate collection of DIY Arduino projects!

In this easy-to-follow book, electronics guru Simon Monk shows you how to create a wide variety of fun and functional gadgets with the Arduino Uno and Leonardo boards. Filled with step-by-step instructions and detailed illustrations, The TAB Book of Arduino Projects: 36 Things to Make with Shields and Proto Shields provides a cost estimate, difficulty level, and list of required components for each project.

\$30.00



30 BeagleBone Black Projects for the Evil Genius

by Christopher Rush

Fiendishly Fun Ways to Use the BeagleBone Black!

This wickedly inventive guide shows you how to program and build fun and fascinating projects with the BeagleBone Black. You'll learn how to connect the BeagleBone Black to your computer and program it, quickly mastering BoneScript and other programming tools so you can get started right away.

\$25.00

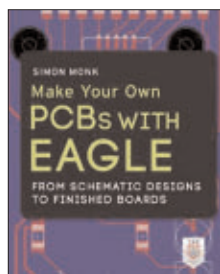


Make Your Own PCBs with EAGLE

by Eric Kleinert

Featuring detailed illustrations and step-by-step instructions, *Make Your Own PCBs with EAGLE* leads you through the process of designing a schematic and transforming it into a PCB layout. You'll then move on to fabrication via the generation of standard Gerber files for submission to a PCB manufacturing service. This practical guide offers an accessible, logical way to learn EAGLE and start producing PCBs as quickly as possible.

\$30.00



Build Your Own Transistor Radios

by Ronald Quan

A Hobbyist's Guide to High Performance and Low-Powered Radio Circuits

Create sophisticated transistor radios that are inexpensive yet highly efficient. Inside this book, it offers complete projects with detailed schematics and insights on how the radios were designed. Learn how to choose components, construct the different types of radios, and troubleshoot your work.

*Paperback, 496 pages

\$49.95

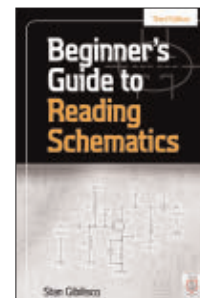


Beginner's Guide to Reading Schematics, 3E

by Stan Gibilisco

Navigate the roadmaps of simple electronic circuits and complex systems with help from an experienced engineer. With all-new art and demo circuits you can build, this hands-on, illustrated guide explains how to understand and create high-precision electronics diagrams. Find out how to identify parts and connections, decipher element ratings, and apply diagram-based information in your own projects.

\$25.00



How to Diagnose and Fix Everything Electronic

by Michael Jay Geier

Master the Art of Electronics Repair

In this hands-on guide, a lifelong electronics repair guru shares his tested techniques and invaluable insights. *How to Diagnose and Fix Everything*

Electronic shows you how to repair and extend the life of all kinds of solid-state devices, from modern digital gadgetry to cherished analog products of yesteryear.

\$24.95



Programming PICs in Basic

by Chuck Hellebuyck

If you wanted to learn how to program microcontrollers, then you've found the right book! Microchip PIC microcontrollers are being designed into electronics throughout the world and none is more popular than the eight-pin version. Now the home hobbyist can create projects with these little microcontrollers using a low cost development tool called the CHIPAXE system and the Basic software language. Chuck Hellebuyck introduces how to use this development setup to build useful projects with an eight-pin PIC12F683 microcontroller.



\$14.95

Programming Arduino Next Steps: Going Further with Sketches

by Simon Monk

In this practical guide, electronics guru Simon Monk takes you under the hood of Arduino and reveals professional programming secrets. Also shows you how to use interrupts, manage memory, program for the Internet, maximize serial communications, perform digital signal processing, and much more. All of the 75+ example sketches featured in the book are available for download.



\$20.00

Order online @ www.store.nutsvolts.com
Or CALL 1-800-783-4624 today!

EDUCATIONAL

Raspberry Three Book Combo



Only \$48.95
Plus
FREE Priority Mail Shipping
US Only

An Arduino Workshop

Are you puzzled about the Arduino but finding it difficult to get all the pieces in one place?



Joe Pardue
SmileyMicros.com **Book and Kit Combo \$124.95**

Arduino Classroom
Arduino 101 Projects Kit



\$44.99
From Smiley's Workshop

CD-ROM SPECIAL

Nuts & Volts
10 CD-ROMs
& Hat Special!

That's 120 issues.
Complete with supporting
code and media files.



Free Shipping!

Only \$208.95
or \$24.95 each.

NUTS & VOLTS
2014
Volume 35, No. 1-12

The 2014 CD-ROM

Pre-Order Special!

Only \$19.95
Free Shipping
Offer Ends December 31st

Published monthly by
T & L Publications, Inc.
www.nutsvolts.com

**GIFTS
FOR
GEEKS**

CALL 1-800-783-4624 today!
Order online @ www.nutsvolts.com

PROJECTS

Accelerometer Kit



The Hockey Puck accelerometer is a compact digital G-force recorder. Find out if your packages are being handled with care (or not!) by your shipping company. See if your crated animals are being well treated on airplane flights. See how many Gs are being applied to the groceries in the trunk of your car. Tie it to the end of a string and see how many Gs you can generate swinging it by hand.

\$59.95

Peek-a-boo Ghost Kit

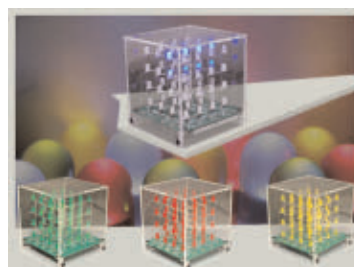


The peek-a-boo ghost kit is a fun, low cost multi-use microcontroller kit. When triggered by the included motion sensor, this mini animatronic waves its arms, lights its LED eyes, and plays back the sounds you record. Perfect for kids, this kit can be used to create a fun Halloween prop for your desk, front door, or walkway. Watch the video to see this cool kit in action. Available in both a program-it-yourself or with a pre-programmed PICAXE chip option.

Un-programmed Chip Kit \$29.95

Pre-programmed Chip Kit \$37.95

3D LED Cube Kit



This kit shows you how to build a really cool 3D cube with a 4 x 4 x 4 monochromatic LED matrix which has a total of 64 LEDs. The preprogrammed microcontroller that includes 29 patterns that will automatically play with a runtime of approximately 6-1/2 minutes. Colors available: Green, Red, Yellow & Blue

\$57.95

Solar Charge Controller Kit 2.0



If you charge batteries using solar panels, then you can't afford not to have them protected from over-charging. This 12 volt/12 amp charge controller is great protection for the money. It is simple to build, ideal for the novice, and no special tools are needed other than a soldering iron and a 9/64" drill!

\$27.95

Geiger Counter Kit



This kit is a great project for high school and university students. The unit detects and displays levels of radiation, and can detect and display dosage levels as low as one micro-roentgen/hr. The LND 712 tube in our kit is capable of measuring alpha, beta, and gamma particles.

Partial kits also available.

\$159.95

Super Detector Circuit Set



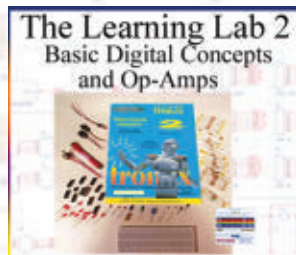
Pick a circuit! With one PCB you have the option of detecting wirelessly: temperature, vibration, light, sound, motion, normally open switch, normally closed switch, any varying resistor input, voltage input, mA input, and tilt, just to name a few.

\$32.95

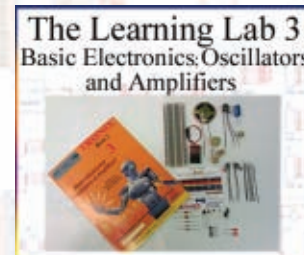
FOR BEGINNER GEEKS!



\$59.95



\$49.95



\$39.95

The labs in this series — from GSS Tech Ed — show simple and interesting experiments and lessons, all done on a solderless circuit board.

As you do each experiment, you learn how basic components work in a circuit, and continue to build your arsenal of knowledge with each successive experiment.

For more info and a promotional video, please visit our webstore.

>>> QUESTIONS

Nixie Tube Specs

I have six F-itron DG10A nixie tubes. I need to find specs and pin-outs for these tubes to see if they are okay. Any info is appreciated.

#12141

Alan Moser
Memphis, TN

Unregulated Power Supply

I picked up a 9V switching power supply wart from one of the online suppliers that advertise in *Nuts & Volts*. Unfortunately, it melted two Arduino's before I realized what was happening. Apparently, the no-load voltage on a switching supply can be up to twice the rated voltage — in my case, 18V. Should I add a load resistor to the internal circuit of the wart so there's always a load? Any other suggestions?

#12142

Fritz Sommer
Knoxville, TN

Photomicrographs

I need photomicrographs of some 3D printed surfaces for a science fair poster. Is there a commercial electron microscope imaging service that you can recommend? I tried my local community college, but they weren't willing to help.

#12143

Fabian Hoffmann
Seattle, WA

What's Up With 3V Sensors?

I'm working with MEMS devices — especially the new 3V sensors. Compared with older 5V sensors, they seem more susceptible to noise. It seems as though dynamic range

would naturally be lower, given the lower operating voltage. Are 3V sensors really a compromise?

#12144

Seymour Holland
Peoria, IL

False Readings

I bought one of those Internet-aware soil moisture devices a few months ago. It worked great at first, but now the electrodes are oxidized and the readings are falsely dry because of the increased resistance. Scrubbing the electrodes with steel wool works for about a week. Any ideas for a permanent solution?

#12145

Harland Foster
Columbus, OH

>>> ANSWERS

D Cell or Gell Cell Adapter for Nikon Cameras

I have the following three Nikon cameras — D3200, D5100, and D5300 — that I want to use in my black bear wildlife research work. There are no electrical outlets to recharge the small lithium-ion batteries that come with the cameras in the desolate areas that I frequent. A car inverter to a regular lithium charger is out of the question because four-wheel drives can't get back to the remote areas that are accessible only by horse and by packing on foot.

I would like to run a bipolar cable from a couple of parallel connected 12 volt/18 amp-hour batteries into a drilled hole in the camera battery snap cover in the camera base. That probably will require cutting open and possibly destroying a battery pack and removing the lithium-ion cells so as to

connect the voltage dropped wires to the battery terminals inside the battery pack which, in turn, will make contact with the camera's internal fixed contacts.

Can you please assist me with a "camera safe" regulated circuit to drop the 12 volt gell cells to a constant regulated output level for a Nikon EN-EL14 lithium-ion battery which is rated at 7.4 volts at 1,050 mAh (7.8 Wh)? Also, battery disassembly details, or if disassembly is not required, how to proceed with the hook-up.

I want to offer some facts that may help you form your own solution.

1 - 7.4 volts happens to be 2X of 3.7 volts, and 3.7 volts happens to be the exact rating of most all cell phone batteries. Cell phone batteries can be bought very cheaply via the Internet.

2 - The next fact is that the battery door on Nikon cameras can be removed. Open it to approx 45 degrees and gently twist it out.

3 - The next fact I would share is that a battery grip is a way to attach (usually) two batteries, to the bottom of your camera. Some of the grips come with interval timers for timed exposures, they are all made in China, and as such, eBay is full of second party battery grips for many popular cameras. The grip might be the better way to modify your cameras without damaging them.

If you were to use cell phone batteries to make your own battery pack, your next challenge would be to build a charger for them, since charging them one by one would be labor intensive. Making a battery pack of the cell batteries would require them to all be the same rated capacity. Connect them in series — in pairs — to bring them up to 7.4 volts and parallel as many as you like to increase capacity to meet your requirements.

Lastly, *NEVER* attempt to open a battery — especially a lithium one. If you were to expose the lithium to air, it most likely will explode and seriously

All questions *AND* answers are submitted by *Nuts & Volts* readers and are intended to promote the exchange of ideas and provide assistance for solving technical problems. All submissions are subject to editing and will be published on a space available basis if deemed suitable by the publisher. Answers are submitted

by readers and **NO GUARANTEES WHATSOEVER** are made by the publisher. The implementation of any answer printed in this column may require varying degrees of technical experience and should only be attempted by qualified individuals.

Always use common sense and good judgment!

injure you or result in your death.

Joseph Massimino
Jensen Beach, FL

[#11142 - November 2014]

Brushless Fans and Solar Panels

I have three 12 VDC brushless fans and am considering running them from a 12V 30W solar panel. Two of the fans are rated at 5.4W and one at 7.6W. Unfortunately, the brushless motors can tolerate a maximum voltage of 13.8V and the solar panel has an open circuit voltage of over 17V. I am afraid I could fry the electronics in the fans with this solar panel but I can't find a 12V solar panel that outputs no more than 12V. I am sure this is not the first time this problem has occurred. What options do I have?

#1 I would use a simple LM7812 voltage regulator for each fan. The 7812 can take up to 18V in and as low as 5V, and has a current rating of one amp. Based on your specs, you will

only be drawing a little over half an amp on the 7.6W fan, and a little under a half amp on the smaller fans. If you find that they get hot, you can easily add a heatsink.

I will also mention that I did something similar, running a boom box radio at the beach, and found that unless the panel is pointed at the sun constantly, the output will drift when fair weather clouds cross overhead. So, I connected three 1000000 20V caps in series, which slowed down the drifting effect caused by clouds.

Another option is to use a solar charger controller like the one I picked up at Harbor Freight Tools. You would hook up the solar panel to the controller, then the controller to the battery (car/marine), and run the fans off the battery. I would still use the LM7812 voltage regulator just to protect the fans.

Dave Little
Lynn, MA

#2 Your solar panels only show that high of voltage when they are unloaded. Once you put the fans on them, the voltage will adjust to their rated voltage, providing you have enough sunlight to drive them properly.

The other good news is that the fans will not burn up when more voltage is applied to them. They will run faster and wear out sooner, but to fry them, you would need much more than a single solar panel can produce.

The solar panel you have says 30 watts, but

what they actually mean is that under ideal conditions, it can produce 30 watts. Two panels would assure that the fan would run in less than ideal sunlight.

Joseph Massimino
Jensen Beach, FL

Carbon or Metal?

Is there a rule of thumb for when it is better to use carbon film resistors over metal film resistors?

Generally, if you require greater precision, better stability, and/or less noise, you go with metal film resistors. In the old days, this was an expensive option, but these days, precision metal film resistors have plummeted in cost — at least for the Asian imports. So, use metal film if you're working with test equipment attenuators, voltage dividers, and analog timing circuits.

Metal Film Is Better:

- If you need a circuit to remain within operational specs for a long time without periodic adjustment.
- If you're designing a low noise audio preamp.
- If you need precise timing for your 555 circuit.

Some notes here:

1 - The ancient trick of beginning with a carbon composition resistor of a lower value than needed and filing a notch in it until the value was *right on*, gave you a good tight resistor for a few minutes. Even if that notch is sealed with clear fingernail polish, carbon resistors still have a lousy temperature coefficient. The resistor will change in value with any changes in temperature far beyond what you'd want with a precision part. It isn't the same as using a precision metal film resistor, no matter what some old timer tells you.

2 - The second note concerns power dissipation. The old carbon comp resistors could take short spikes in power and survive even with short bursts of ten times the power rating because the mass of that slug of a resistance element could absorb and dissipate that spike. Carbon film and



AP CIRCUITS
PCB Fabrication Since 1984

As low as...

\$9.95
each!

Two Boards
Two Layers
Two Masks
One Legend

Unmasked boards ship next day!

www.apcircuits.com



Send all questions and answers by email to forum@nutsvolts.com
or via the online form at www.nutsvolts.com/tech-forum

metal film resistors can handle their specified power level, but can't handle spikes well because they don't have that same mass in the resistance element. They can dissipate the average rated power, but not absorb the heat of a big power spike. So, that same spike in power can cause a carbon or metal film resistor to literally burn out.

Even with the problem of film

over composition described above, carbon film resistors do seem to have tighter and more stable values than the comps. They still suffer from bad temperature coefficients and noise, but the typical new 5% carbon film resistors will often be well within 2% of its marked value.

**Dean Huster
Harviell, MO**

Editor's Note —

Dean Huster passed away on October 12, 2014. Dean was a long time contributor to this column and also participated regularly in our online forums. Many will remember his contribution to the hobbyist community as the Q&A guy for Poptronics Magazine before they closed up shop. He will be missed.

■ LOOK FOR ■ SEARCH FOR ■ FIND

Find your favorite advertisers here!

ADvertiser INDEX

AMATEUR RADIO

AND TV

National RF25
Ramsey Electronics82-83

BUYING ELECTRONIC SURPLUS

Earth Computer Technologies16
Weirdstuff Warehouse25

CCD CAMERAS/VIDEO

Ramsey Electronics82-83

CIRCUIT BOARDS

Accutrace Inc..17
AP Circuits80
Digilent4
Dimension Engineering74
ExpressPCB23
Front Panel Express LLC8
Little Bits Electronics... Back Cover
PCB Pool8
Saelig Co. Inc.31

COMPUTER

Hardware

Earth Computer Technologies16
Weirdstuff Warehouse25

Microcontrollers / I/O Boards

M.E. Labs24
MikroElektronika3
Parallax, Inc.31
Technologic Systems59

DESIGN/ENGINEERING/ REPAIR SERVICES

Accutrace Inc..17

Cleveland Institute of Electronics..65
ExpressPCB23
Front Panel Express LLC8
National RF25
PCB Pool8

EDUCATION

Boxed Kit Amps25
Cleveland Institute of Electronics..65
Command Productions9
Digilent4
Little Bits Electronics... Back Cover
Lumidax15
NKC Electronics25
Poscope.....13

EMBEDDED TOOLS

NetBurner2

ENCLOSURES

Front Panel Express LLC8

HI-FI AUDIO

Boxed Kit Amps25

KITS & PLANS

Boxed Kit Amps25
Earth Computer Technologies16
NetBurner2
NKC Electronics25
Ramsey Electronics82-83

MISC./SURPLUS

All Electronics Corp.71
Front Panel Express LLC8
Weirdstuff Warehouse25

MOTORS

Servo City/Robot Zone37

PCB ASSEMBLY

Accutrace Inc..17

PROGRAMMERS

Little Bits Electronics... Back Cover
M.E. Labs24
MikroElektronika3

RF TRANSMITTERS/ RECEIVERS

National RF25

ROBOTICS

Cleveland Institute of Electronics..65
Digilent4
GoSphero5
Servo City/Robot Zone37

TEST EQUIPMENT

Dimension Engineering74
Lumidax15
NKC Electronics25
Pico Technology7
Poscope.....13
Saelig Co. Inc.31

TOOLS

MikroElektronika3
NetBurner2
PanaVise16
Poscope.....13

VIDEO/DISPLAY

MODULES
4D Systems43

WIRE, CABLE AND CONNECTORS

Servo City/Robot Zone37

4D Systems43
Accutrace Inc..17
All Electronics Corp.71
AP Circuits80
Boxed Kit Amps25
Cleveland Institute of
Electronics65
Command Productions9
Digilent4
Dimension Engineering.....74
Earth Computer Technologies 16
ExpressPCB23
Front Panel Express LLC8
GoSphero5
Little Bits
ElectronicsBack Cover
Lumidax15
M.E. Labs.....24
MikroElektronika3
National RF25
NetBurner2
NKC Electronics25
PanaVise16
Parallax, Inc.31
PCB Pool8
Pico Technology7
Poscope.....13
Ramsey Electronics82-83
Saelig Co. Inc.31
Servo City/Robot Zone37
Technologic Systems59
Weirdstuff Warehouse25



Get Ready For The Holidays With a Ramsey Kit!



Super-Pro FM Stereo Radio Station Kit

- ✓ PLL synthesized for drift-free operation
- ✓ Built-in mixer - 2 line inputs and one microphone input, line level monitor output!
- ✓ Frequency range 88.0 to 108.0, 100 kHz steps
- ✓ Precision active low-pass "brick wall" audio filter!
- ✓ Dual LED bar graph audio level meters!
- ✓ Automatic adjustable microphone ducking!
- ✓ Easy to build through-hole design!

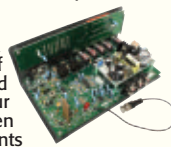
FALL FM SALE!



The true professional workhorse of our FM Stereo transmitter line, the FM100B has become the transmitter of choice for both amateurs and professionals around the world. From the serious hobbyist to churches, drive-in theaters, colleges and schools, it continues to be the leader. Not just a transmitter, the FM100B is a fully functional radio station and provides everything but the audio input and antenna system! Just add that and you're on the air!

This professional synthesized transmitter is adjustable directly from the front panel with a large LED digital read-out of the operating frequency. Just enter the setup mode and set your frequency. Once selected and locked you are assured of a rock stable carrier with zero drift. The power output is continuously adjustable throughout the power range of the model selected. In addition, a new layer of anti-static protection for the final RF amplifier stage and audio inputs has been added to protect you from sudden static and power surges.

Audio quality is equally impressive. A precision active low-pass brick wall audio filter and peak level limiters give your signal maximum "punch" while preventing overmodulation. Two sets of rear panel stereo line level inputs are provided with front panel level control for both. Standard unbalanced "RCA" line inputs are used to make it simple to connect to the audio output of your computer, MP3 player, DVD player, cassette deck or any other consumer audio source. Get even more creative and use our K8094 below for digital storage and playback of short announcements and ID's! In addition to the line level inputs, there is a separate front panel microphone input.



All three inputs have independent level controls eliminating the need for a separate audio mixer! Just pot-up the source control when ready, and cross fade to the 2nd line input or mic! It's that simple! In addition to the dual stereo line inputs, a stereo monitor output is provided. This is perfect to drive studio monitors or local in-house PA systems. The FM100B series includes an attractive metal case, whip antenna and built in 110/220VAC power supply. A BNC connector is also provided for an external antenna. Check out our Tru-Match FM antenna kit, for the perfect mate to the FM100B transmitter. We also offer a high power kit as well as an export-only assembled version that provides a variable RF output power up to 1 watt. The 1 watt unit must utilize an external antenna properly matched to the operating frequency to maintain a proper VSWR to protect the transmitter.

(Note: The FM100B and FM100BEX are do-it-yourself learning kits that you assemble. The end user is responsible for complying with all FCC rules & regulations within the US or any regulations of their respective governing body. The FM100BWT is for export use and can only be shipped to locations outside the continental US, valid APO/FPO addresses or valid customs brokers for documented end delivery outside the continental US.)

FM100B Super-Pro FM Stereo Radio Station Kit, 5uW to 25mW Output
FM100BEX Super-Pro FM Stereo Radio Station Kit, 5uW to 1W Output

FALL FM SALE!
\$239.95
\$299.95

Digital Controlled FM Stereo Transmitters

- ✓ PLL synthesized for drift free operation
- ✓ Front panel digital control and display of all settings and parameters!
- ✓ Professional metal case for noise-free operation
- ✓ EMI filtering on audio and power inputs
- ✓ Super audio quality, rivals commercial broadcasts
- ✓ Available in domestic kit or factory assembled export versions

FALL FM SALE!



For more than a decade we've been the leader in hobbyist FM radio transmitters. We told our engineers we wanted a new technology transmitter that would provide FM100 series quality without the advanced mixer features. They took it as a challenge and designed not one, but TWO transmitters!

The FM30B is designed using through-hole technology and components and is available only as a do-it-yourself kit with a 25mW output very similar to our FM25 series. Then the engineers redesigned their brand-new design using surface mount technology (SMT) for a very special factory assembled and tested FM35BWT version for our export only market! All settings can be changed without taking the cover off! Enter the setup mode from the front panel and step through the menu to make all of your adjustments. A two line LCD display shows you all the settings! In addition to the LCD display, a front panel LED indicates PLL lock so you know you are transmitting.

Besides frequency selection, front panel control and display gives you 256 steps of audio volume (left and right combined) as well as RF output power. A separate balance setting compensates for left/right differences in audio level. In addition to settings, the LCD display shows you "Quality of Signal" to help you set your levels for optimum sound quality. And of course, all settings are stored in non-volatile memory for future use! Both the FM30B and FM35BWT operate on 13.8 to 16VDC and include a 15VDC plug-in power supply. (Note: After assembly of this do-it-yourself hobby kit, the user is responsible for complying with all FCC rules & regulations within the US, or any regulations of their respective governing body. FM35BWT is for export use and can only be shipped to locations outside the continental US or valid APO/FPO addresses or valid customs brokers for end delivery outside the continental US.)

FM30B Digital FM Stereo Transmitter Kit, 0-25mW, Black
FM35BWT Digital FM Stereo Transmitter, Assembled, 0-1W, Black (Export ONLY)

FALL FM SALE!
\$169.95
\$259.95



There's only so much room on these two pages, so check it all out in our new virtual electronic catalog! Flip through the pages and search with ease! Visit www.ramseykits.com

The Holiday Spotlight!

LED Animated Santa



This animated Santa and reindeer display has been our most popular holiday display for years! It contains a whopping 126 dazzling colored LEDs which make it a great holiday sign that is guaranteed to draw attention!

LED animated motion makes it come alive. Runs on standard 9V battery or 9-12VDC external power supply. Dazzle your friends this great display!

MK116 LED Animated Santa Kit \$21.95

LED Christmas Tree



Electronic Christmas tree features 134 bright colored LEDs in the shape of a gorgeous holiday Christmas tree. Includes 18 random flashing blinking "candles" on the PC board! Runs on a 9V battery or external 9-12VDC power supply.

MK117 LED Christmas Tree Kit \$24.95

LED Animated Holiday Bell



This PC board holiday bell is animated to simulate a bell swinging back and forth! 84 bright colored LEDs will dazzle you with holiday cheer! Includes an on/off switch. Runs on 9V.

MK122 LED Animated Bell Kit \$18.95

3D LED Christmas Tree



Not your average LED display! 4 branch sections give this tree a true 3D look! 16 red LEDs light it up with yellow LEDs for you to customize your tree! The base of the tree is actually the 9V battery acting as a self supporting base! Now that's pretty neat!

MK130 3D LED Christmas Tree Kit \$7.95

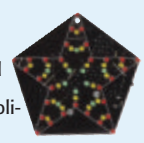
SMT LED Christmas Tree



Build this subminiature Christmas tree and learn SMT at the same time. Small enough to wear as a badge or pendant! Extra SMT parts are included so you can't go wrong! Runs on Li-Ion cell.

MK142 SMT LED Christmas Tree Kit \$10.95

Sound Activated LED Star



A built-in microphone picks up music and room audio and the LEDs respond just like a professional LED Vu meter! Adjustable sensitivity creates a great holiday display in sync with your music or audio! Runs on 9-12VDC.

MK172 Sound Activated LED Star Kit \$19.95

LED Flashing Holiday Star



A classic holiday star shaped PC board contains 35 brilliant red or yellow LEDs that can be selected to provide a steady or flashing display. The built-in 9VDC battery holder acts as the base for the star, making it easy to add to your decorations!

MK169 LED Flashing Holiday Star \$12.95

LED Animated Effects Star



The ultimate LED star display features 60 brilliant LEDs that are micro-processor controlled to provide 24 different effects with sequencing! Includes red and yellow LEDs. Runs on a standard

MK170 LED Animated Star Kit \$29.95

Follow Us and SAVE \$\$

Follow us on your favorite network site and look for a lot of super deals posted frequently... exclusively for our followers!



Next Generation Classic Nixie Tube Clocks!



- ✓ Not your ordinary holiday clock gift!
- ✓ Today's latest technology with yesterday's display!
- ✓ Choice of hand rubbed Teak/Maple or acrylic base!
- ✓ Choice of Nixie tubes types
- ✓ Low-cost GPS receiver timebase option!
- ✓ Programmable color LED tube mood lighting!
- ✓ The ultimate conversation piece!

THE NEW STANDARD IN NIXIE CLOCKS!

Our next generation of classic Nixie tube clocks perfectly mesh today's technology with the Nixie era technology of the 60's. Of course, features you'd expect with a typical clock are all supported with the Nixie clock... and a whole lot more! Time wise, the clocks are designed around a quartz crystal timebase, therefore are not AC power frequency dependent like a lot of clocks. This means they can be used in any country regardless of power frequency, with the included 12VDC regulated power supply. The clocks are also programmable for 12 or 24 hour mode, various AM/PM indications, programmable leading zero blanking, and include a programmable alarm with snooze.

Unlike most Nixie clocks, the clocks also display the date in DD.MM.YY, MM.DD.YY, or YY.MM.DD format, which can be programmed to display for a few seconds at the end of each minute either as a static display, display wise, the clocks feature a programmable night mode with dim or blank display, a programmable master blank tube saver, hard or soft fade digit change, and even have a built-in "Slot Machine" cathode poisoning prevention routine. Programming and setting the clock is a breeze with simple 2-button entries on the rear panel. The clocks are available in our signature hand rubbed Teak & Maple or futuristic clear acrylic bases.

WE CROSSED THE TECHNOLOGY TIMELINE!

We then jumped the technological timeline of the 60's Nixie displays by adding the latest multi-colored LEDs to the base of the Nixie tubes to provide hundreds of illumination colors to highlight the glass tubes! The LED lighting can be programmed to any color and brightness combination of the colors red, green, or blue to suit your mood or environment. Then we leaped over the technological timeline by integrating an optional GPS time base reference for the ultimate in clock accuracy! The small optional GPS receiver module is factory assembled and tested, and plugs directly into the back of the clock to give your Nixie clock accuracy you could only dream of! The new series clocks are available in 6-tube and 4-tube versions, with your choice of bases, and your choice of kit or factory assembled & tested. If you're looking for the ultimate conversation piece, with a trip down nostalgia lane, check out our clocks at www.ramseykits.com.

NIXIE CLOCKS

Classic Nixie Tube Clocks, Teak/Maple or See-Through Acrylic Base, Kit or Factory Assembled

From \$229.95

The Learning Center!

- ✓ Learn and build!
- ✓ 130, 200, 300, & 500 in one electronic labs!
- ✓ Practical through hole and SMT soldering labs!
- ✓ Integrated circuit AM/FM radio lab!
- ✓ Beginner's non-soldering & soldering kits!

130-In-One Learning Lab

Learn the basics and theory of electronics the fun way while building 130+ projects that you actually see working when you're done! The detailed instruction manual covers the hows and whys of every circuit, theory, parts identification, and description.

PL130A 130-In-One Learning Lab \$39.95

200-In-One Super Fun Lab

The front panel contains a built-in speaker, earphone, meter, 7-segment LED digital display, two controls and more to make the finished projects fun to use. Now you can build your own AM broadcast station, burglar alarm, telegraph, sound effects, radios, and more!

PL200 200-In-One Learning Lab \$84.95

300-In-One Digital Lab

Take that next step and jump up to 300 separate projects that start walking you through the learning phase of digital electronics. This learning lab along with its comprehensive manual makes digital electronics learning fun!

PL300 300-In-One Learning Lab \$109.95

500-In-One Ultimate Lab

The ultimate electronics lab that includes 500 separate projects that cover it all, from the basics all the way to digital programming. Allows you to start at the beginning and progress at your own speed with 500 projects! Heavy duty case!

PL500 500-In-One Ultimate Lab \$249.95

Practical Soldering Lab

Whether young or old, there's always a need to hone your soldering skills. Either learn from scratch or consider it a refresher, and end up with a neat little project when you're done! Covers through-hole design & soldering.

SP1A Practical Soldering Lab \$9.95

SMT Soldering Lab

Includes the PC board, all the SMT and other components required, and a comprehensive training course covering all aspects of SMT soldering, desoldering, removing, and component placement. The finished project is a neat little "Decision Maker".

SM200K SMT Soldering Lab \$22.95

AM/FM Radio Lab

The unique design includes an IC, transistors, resistors, capacitors, coils, and misc. parts. A soldering guide is also provided. The radio kit includes a hi-tech blue PC board with the easy to follow schematic printed on the top surface.

AMFM108K AM/FM Radio Lab \$36.95

RC Race Car Learning Kit

One of the neatest and most exciting learning kits for the kids! You will be building the Turbo King R/C speedster from the ground up, ending up with a super fast car! At the same time learning transmitters, receivers, switches, gears, and motors!

AK870 R/C Car Learning Kit \$29.95

GET THE INUTS AND VOLTS DISCOUNT!

Mention or enter the coupon code **NVRM142** and receive **10% off your order!**

800-446-2295
www.ramseykits.com

Prices, availability, and specifications are subject to change. We are not responsible for typos, stupid, printer's bleed, or watching too much football on TV! Robin gave me 4 days to get this ad copy done... Nice Holiday Spirit! Visit www.ramseykits.com for the latest pricing, specials, terms and conditions. Copyright 2014 Ramsey Electronics® ... so there!



RAMSEY ELECTRONICS®

590 Fishers Station Drive
Victor, NY 14564
(800) 446-2295
(585) 924-4560



littleBits™
ELECTRONICS

the bitLab™

THE APP STORE FOR HARDWARE IS HERE.

Get started at

littleBits.cc/bitLab

littleBits makes an expanding library of modular electronics that snap together with magnets. It's the easiest and most extensive way to learn and prototype with electronics. Create complex circuits in seconds: no soldering, wiring, or programming required (unless you want to).

The bitLab is the new marketplace for user-generated hardware. You can submit your own hardware designs for modules you want to see added to the littleBits library. If your design gets enough community support, we'll manufacture it, market it, sell it and give you a percentage of the sales!